SYBEX COMPUTER BOOKS

ATARI®

BASIC PROGRAMS IN MINUTES

STANLEY R. TROST

ATARI

BASIC PROGRAMS IN MINUTES

ATARI™
BASIC PROGRAMS IN MINUTES

STANLEY R. TROST

BERKELEY • PARIS • DÜSSELDORF

Cover art by Jean-François Pénichoux
Design by Nina Hoecker

Every effort has been made to supply complete and accurate information. However, SYBEX assumes no responsibility for its use, nor for any infringements of patents or other rights of third parties which would result.

Library of Congress Card Number: 83-51192
ISBN 0-89588-154-3
Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

ACKNOWLEDGEMENTS

Many people contributed ideas and hard work to this book. Rudolph Langer suggested the book and pointed the way to a number of improvements. David Lichtblau provided valuable assistance with program development and verification. Barbara Gordon and Joel Kreisman edited the manuscript, checked the programs, and prepared the programs for production. Many other members of the SYBEX staff were also part of the team that put this book together.

Stan Trost
August 1983

CONTENTS

APPENDICES

PREFACE

Atari BASIC Programs in Minutes will help you discover the full power and flexibility of your Atari computer. This book contains more than 65 practical programs and subroutines that are fully tested and ready to run; they provide a wide variety of business, personal, and educational applications. You do not need to understand the BASIC programming language to use these programs; you just enter them into your computer and use them for your own applications.

These programs will help you make important financial decisions; for example, you can find the break-even point of a new business, calculate the monthly payment necessary to establish a college fund for your children, and find the future value of regular deposits in an individual retirement account (IRA). The record keeping programs allow you to create a mailing list, phone directory, and automobile log. Students and nonstudents alike will find the mathematics practice programs especially useful. These practice programs help sharpen arithmetic skills; they can also help students with homework assignments.

Atari BASIC Programs in Minutes contains a number of unique programs. The fraction practice programs, long multiplication drill, record keeping programs, and combined data analysis and graphing programs are seldom found in a collection of programs.

The use of standard subroutines is also unique. A primary goal of this book is to provide programs that are *short* and easy to enter in your computer. To achieve this goal, the programs were created using a standard set of programs as subroutines. These subroutines will also save you a great deal of effort when you create your own programs.

This book does not teach BASIC language programming; however, use of the programs will help you understand BASIC in a painless, almost automatic way. As you become familiar with BASIC, you will be able to modify the programs in this book and create new programs.

Let's now review the contents of the book. Chapter 1 gives instructions for using the programs and reviews the interactive features built into them. You should read this chapter before you use the programs.

Chapter 2 is a set of programs for home and business finance. It includes programs that calculate the compound interest you earn in a bank, the future value of regular savings deposits (for example, the future value of deposits in an IRA), the interest rate earned on an investment, and the time required to double your money. These programs can help you make many important investment decisions.

Chapter 3 contains a collection of business programs. These programs compute depreciation, the break-even point for a new business, wages earned with overtime, and decision analysis. The business programs will be helpful to business people, employees, and consumers.

In Chapter 4 we present real estate application programs including mortgage payment calculation, balloon payment calculation, and rental property analysis. You will, for example, be able to find the interest you can save by increasing your monthly house payment. Investors, homeowners, and prospective homeowners will find these programs helpful.

Chapter 5 includes a set of data analysis programs that can be used to analyze both scientific and financial data. The programs can compute mean and standard deviation, various moving averages, and linear regression, and they can graph the results. These programs can be of use in analyzing and projecting stock market trends, among other things.

Chapter 6 contains a set of record keeping programs to keep personal and automobile records. With these programs you will be able to use your computer to keep mailing lists and phone directories as well as mileage records for your car. The programs are designed to be modified easily for other applications.

Chapter 7 presents a number of useful arithmetic drills for students of all grade levels and abilities. These drills include addition, subtraction, multiplication, and division, using whole numbers as well as fractions. These programs can help sharpen your math skills and even help with homework!

We also include two appendices with this book. Appendix A lists the standard subroutines that enable the programs within the chapters to be so short. Their functions are described in REM statements. These subroutines are in numerical order so that you can easily refer to them. Appendix B shows you how to use these subroutines to create your own programs. Descriptions and examples are provided.

1
INTRODUCTION

This chapter shows you how *Atari BASIC Programs in Minutes* can greatly increase your computer's effectiveness. You will learn how to enter and run programs, create a subroutine library, and build a library of related programs that run from a menu. You will also learn what interactive program design means and how it is accomplished in these programs.

The programs in this book do not require knowledge of the BASIC programming language. They can simply be used as is to solve a wide variety of business, home, and education problems. However, if you wish to expand your programming knowledge, these programs will show you how to take full advantage of the powerful features of your computer.

HOW TO START

You will need an Atari computer with at least 48K of memory and a disk drive or tape cassette. (You need the disk drive to run the programs in Chapter 6.) If you have an Atari 800 or 1200XL you will also need a BASIC program cartridge. The programs in this book are written in Atari BASIC. Review your BASIC manual for detailed instructions on loading and running BASIC.

Be sure the BASIC cartridge is installed in the left cartridge slot. If you're using the disk system, turn the disk drive on. When the whirring stops, insert your master diskette. Now turn on the computer. After a short time the screen displays these lines:

```
READY
■
```

The ■ is called the cursor.

You will need two cassette tapes or two formatted disks (formatting instructions are in the Atari DOS manual) to save the programs you enter (one as a master and one as a backup). You should always make two copies of every program.

If you don't have a disk drive, install the BASIC cartridge if necessary, and turn on the computer. When you want to save your programs on tape, follow the Atari instructions for using a tape cassette.

At this point you can enter any of the programs in this book. You enter a program by typing the lines in the program listings exactly as they appear in this book. After each line press the return key. Let's enter the following program:

```
100 PRINT "HELLO"
110 END
```

Begin by typing the first line (100 PRINT "HELLO"). Enter this line by striking the return key. Complete the program entry by typing and entering the second line.

You will sometimes make mistakes in entering the programs. Short lines are easily corrected by retyping the entire line. The Atari BASIC manual explains other methods you can use to correct errors on longer lines.

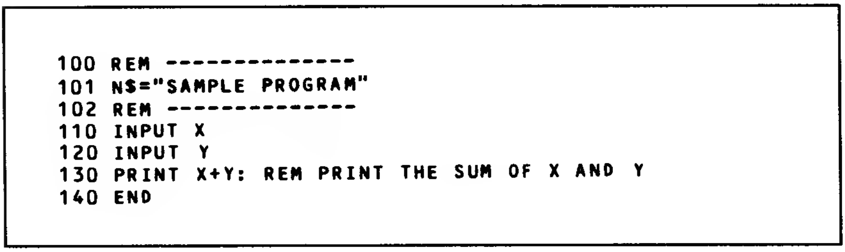
To run this program, type RUN and strike the return key. Notice the display on your screen; your program has printed HELLO below the program you entered. We'll give you more information on entering, saving, and running the programs later in this chapter.

REMARK STATEMENTS

We use *remarks* to make the programs easier to understand. Remark statements begin with the symbol REM. In Figure 1.1, remarks make a border at the top of the program and explain the action of line 130. With one exception, you don't have to type the remarks as you enter the programs. The exception is that you must type the word REM on the first line of the program. The first REM is necessary because of the design of our programs. Figure 1.2 shows the same program listing after you type the necessary lines.

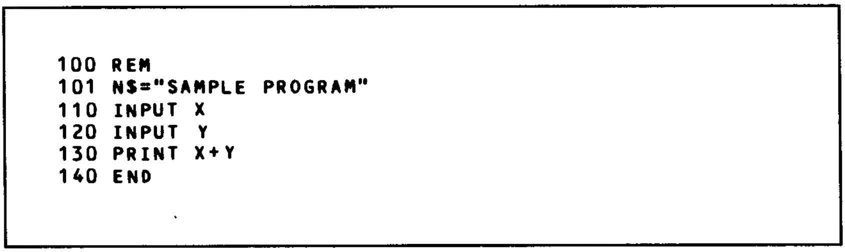
BUILDING A SUBROUTINE LIBRARY

The programs in this book use a central subroutine library to simplify program entry and output formatting. A *subroutine* is simply a program



```
100 REM -------------
101 N$="SAMPLE PROGRAM"
102 REM -------------
110 INPUT X
120 INPUT Y
130 PRINT X+Y: REM PRINT THE SUM OF X AND Y
140 END
```

Figure 1.1: Program Listing Including Remarks



```
100 REM
101 N$="SAMPLE PROGRAM"
110 INPUT X
120 INPUT Y
130 PRINT X+Y
140 END
```

Figure 1.2: Program Listing with Remarks Condensed

that is used as part of another program. For example, one central subroutine handles data entry for many of our programs. These subroutines allow our programs to be short and therefore easy for you to enter.

Appendix A contains the program listings for the subroutines you will use in this book. Appendix B shows you how to use some of these subroutines to create your own programs.

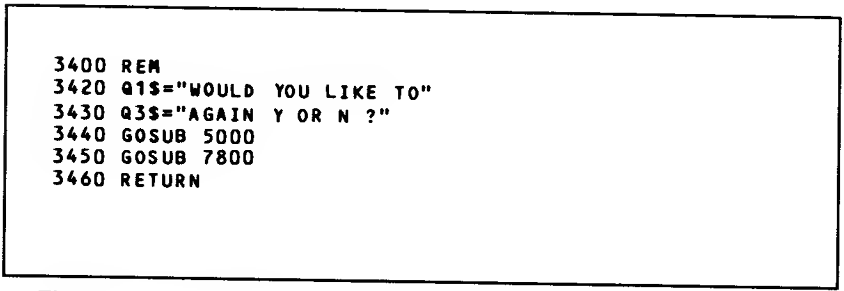
The following list shows the subroutines you must enter for the different chapters in this book. You should enter and save the subroutines on disk as you need them; thus you need enter each subroutine only once. The numbers in the list refer to the first line number of each subroutine.

Chapters 2-7	3400, 5000, 5400, 5800, 6200, 6600, 7000, 7400, 7800
Chapter 6	3000, 3800, 4600

We will begin by building the subroutine library for Chapters 2-5. Turn to Appendix A, where you will find the listings for the standard subroutines. Because the subroutine at line 3000 isn't needed until Chapter 6, begin with line 3400. First type NEW to clear any old programs from the computer; then type the instructions for this subroutine into your computer. (Remember that remarks are not entered except in the first line.) Your entry should match that shown in Figure 1.3.

Continue to enter the rest of the subroutines necessary for Chapters 2-5 and 7. After you enter the standard routines, save them on disk or tape. Follow the instructions in the DOS manual to ready your disk. Then type

```
LIST "D:SUBLIB
```



```
3400 REM
3420 Q1$="WOULD YOU LIKE TO"
3430 Q3$="AGAIN Y OR N ?"
3440 GOSUB 5000
3450 GOSUB 7800
3460 RETURN
```

Figure 1.3: Program Listing: ANOTH Subroutine

To save the subroutines on tape, ready your recorder and type

```
LIST "C:SUBLIB
```

Your central subroutine library is now stored with the file name SUBLIB. You will merge SUBLIB with your application programs when you run the applications. You should add the remainder of the subroutines to SUBLIB when you enter the programs in Chapter 6.

ENTERING AND RUNNING THE PROGRAMS

You are now ready to enter a program. Type the program instructions as they are shown in the program listings. The programs in Chapters 2, 3, 4, and 5 should be saved in one file for each chapter. (Use file names that are meaningful to you.) The programs can then be run by loading the subroutine library, merging the file for the desired chapter, and typing RUN. As you type in the programs in a chapter, you will create a menu to run the programs in that chapter. A *menu* is a list of programs that you use to select the program you want to run. Instructions for creating the menu are given in each chapter. The programs within each chapter begin with unique line numbers; this allows you to store the programs and menu for each chapter together in one file.

For example, Chapter 2 contains a number of programs that are useful when you make financial calculations. You should first type in the Financial Programs Menu and save it on disk or tape by typing

```
LIST "D:FIN
```

(for disk), or

```
LIST "C:FIN
```

(for tape).

Then load the menu you saved by typing

```
ENTER "D:FIN
```

or

```
ENTER "C:FIN
```

Now you can type in another financial program, for example, the

Future Value of a Deposit program. Save both programs together by typing

```
LIST "D:FIN
```

or

```
LIST "C:FIN
```

You can see how this method enables you to build a library of financial programs.

You merge the subroutines and application programs with the following procedure. First clear memory by typing NEW. Now load the subroutines (in this example we are using the disk system):

```
ENTER "D:SUBLIB
```

The subroutines and application programs are automatically merged when you load the application programs:

```
ENTER "D:FIN
```

If you are using a cassette recorder, replace the D with a C.

The Financial Programs Menu enables you to run any program in that chapter by selecting it from the menu. After you have run a program (and chosen not to run it any more), you will be automatically returned to the menu to make another selection.

INTERACTIVE FEATURES

The word *interactive* refers to the interaction between the computer, the programs, and the user. The goal of an interactive program is to make it easy for the user to interact with the computer.

We accomplish this goal in various ways. When possible, we answer questions with a single keystroke that doesn't require the return key. For example, programs are selected from the menus with a single digit, and yes/no questions require simply a Y or an N. Another interactive feature we use is to place all questions at the bottom of the screen, so you can always look for them in the same place. Your answers are listed at the top of the screen, so you can keep track of the data you have input. We also use sound to alert you to possible errors in input, particularly to incorrect answers in the arithmetic drills.

SUMMARY

This chapter provides you with detailed instructions for creating and saving a subroutine library and for using the library with the rest of the programs in this book. This chapter also tells you how to create a library of related programs that can be run from a menu. Each time you need a new program, you add it to the programs you have already saved.

The use of the central subroutine library allows the programs in this book to be very short and easy to enter, and the interactive features built into the programs make them easy to use.

2
HOME AND BUSINESS FINANCE

This chapter contains ten programs useful for many home and business finance applications. For example, let's assume that you are going to deposit money in a bank account for a specific term. One bank pays 8% compounded monthly, and another bank pays 8¼% compounded quarterly. With these programs you can use your computer to find which bank account pays more interest.

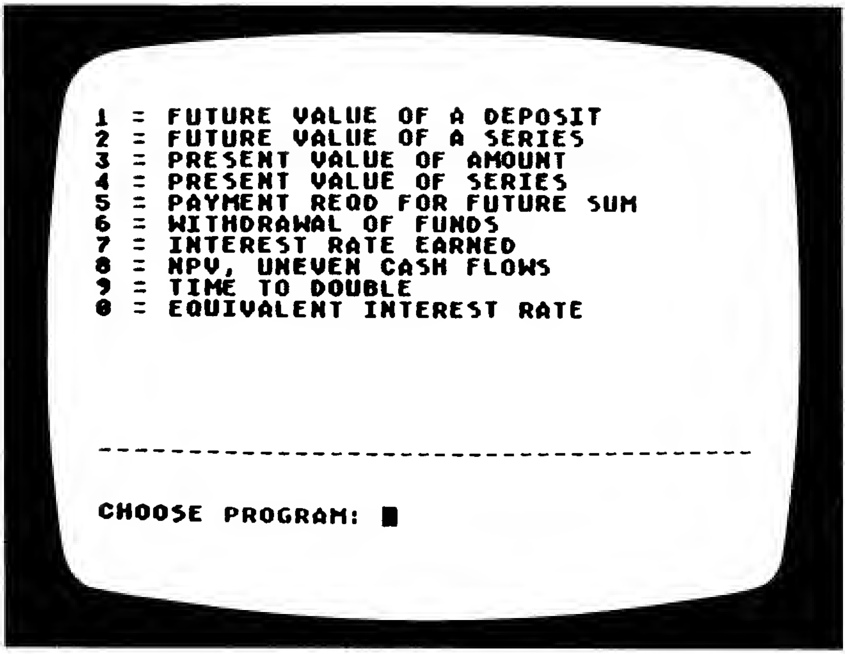
You may also wish to contribute regularly to an individual retirement account (IRA). With one of the programs in this chapter you can compute how much money you will have in your IRA at the time you retire. Another program allows you to see if an amount in a savings account is enough for you to retire with.

FINANCIAL PROGRAMS MENU

The Financial Programs Menu lets you choose one program and run it. The menu is set up so that you press a single key to select the program you want to run. Once you have entered and saved the menu, you can then add the other programs in this chapter one at a time, as you need them. Review chapter 1 to see how to save all the programs in a chapter in a single file.

After you have run the financial programs you need, you can return to BASIC by pressing the reset key—the cursor will appear on a blank line, and you can then load another set of programs.

The menu is shown in Figure 2.1. Figure 2.2 shows the menu selection program.



```
1 = FUTURE VALUE OF A DEPOSIT
2 = FUTURE VALUE OF A SERIES
3 = PRESENT VALUE OF AMOUNT
4 = PRESENT VALUE OF SERIES
5 = PAYMENT REQD FOR FUTURE SUM
6 = WITHDRAWAL OF FUNDS
7 = INTEREST RATE EARNED
8 = NPV, UNEVEN CASH FLOWS
9 = TIME TO DOUBLE
0 = EQUIVALENT INTEREST RATE




--------------------------------------

CHOOSE PROGRAM: █
```

Figure 2.1: Screen Display: Financial Programs Menu

```
100 REM ----------------------------
110 REM FINANCIAL PROGRAMS
120 REM ----------------------------
130 GOSUB 6200:REM INITIALIZE
140 REM SET UP MENU ARRAY
150 X1$="FUTURE VALUE OF A DEPOSIT"
160 X2$="FUTURE VALUE OF A SERIES"
170 X3$="PRESENT VALUE OF AMOUNT"
180 X4$="PRESENT VALUE OF SERIES"
190 X5$="PAYMENT REQD FOR FUTURE SUM"
200 X6$="WITHDRAWAL OF FUNDS"
210 X7$="INTEREST RATE EARNED"
220 X8$="NPV, UNEVEN CASH FLOWS"
230 X9$="TIME TO DOUBLE"
240 X0$="EQUIVALENT INTEREST RATE"
250 N=10:GOSUB 7000:REM DISPLAY MENU
260 REM SELECT PROGRAM
270 ON X GOSUB 290,520,730,910,1110,1310,1580,1770,2040,2320
280 GOTO 150
```

Figure 2.2: Program Listing: Financial Programs Menu

FUTURE VALUE OF A DEPOSIT

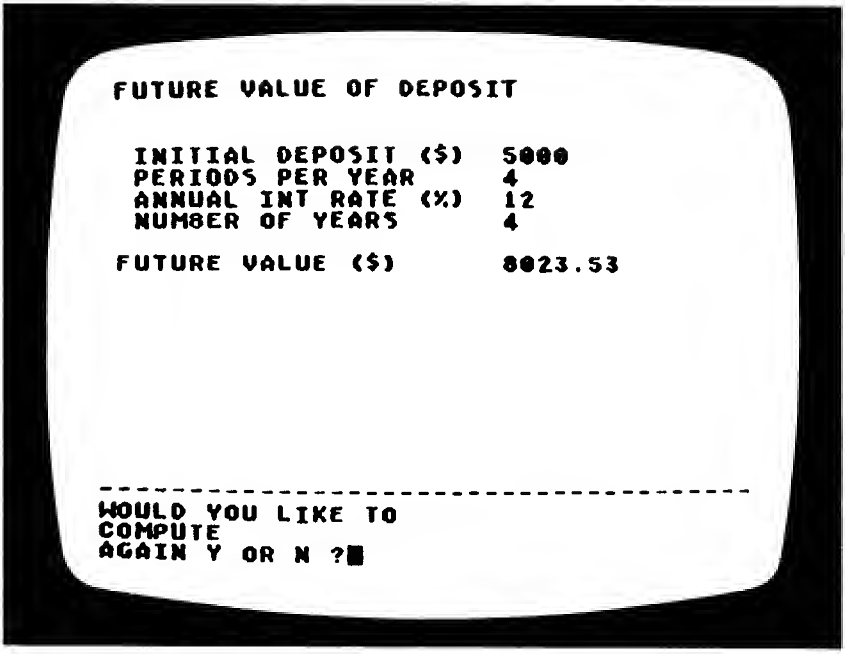
Description

Have you ever wondered how quickly a savings account will grow? Would you like to compare the effect of different interest rates and compounding periods?

You can use this program to compute the future value of a deposit. You must specify the amount of the deposit, the number of times per year interest is compounded, the annual interest rate, and the number of years the funds will earn interest. The program assumes a constant interest rate.

Example

Tom deposits $5,000 in a four-year savings account. The bank pays interest at the rate of 12% per annum, compounded quarterly. Tom wishes to know how much his savings will grow in four years. Figure 2.3 shows the results of the computation on the screen. Figure 2.4 is the program listing for the Future Value program.



```
FUTURE VALUE OF DEPOSIT

 INITIAL DEPOSIT ($)   5000
 PERIODS PER YEAR      4
 ANNUAL INT RATE (%)   12
 NUMBER OF YEARS       4

FUTURE VALUE ($)       8023.53







---------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN Y OR N ?█
```

Figure 2.3: Screen Display: Future Value of a Deposit

```
290 REM ------------------------------
300 N$="FUTURE VALUE OF DEPOSIT"
310 REM ------------------------------
320 GRAPHICS 0:PRINT N$:PRINT
330 K=1:REM INITIALIZE
340 REM ASK FOR PARAMETERS
350 Q1$="ENTER DATA":Q2$=""
360 Q3$="INITIAL DEPOSIT ($)":GOSUB 6600
370 Q3$="PERIODS PER YEAR":GOSUB 6600
380 Q3$="ANNUAL INT RATE (%)":GOSUB 6600
390 Q3$="NUMBER OF YEARS":GOSUB 6600
400 REM DO THE COMPUTATION
410 PRINT
420 PRINT "FUTURE VALUE ($)";:POSITION 24,8
430 A=P(1)
440 FOR I=1 TO P(2)*P(4)
450 A=A+A*P(3)/100/P(2)
460 NEXT I
470 A=INT(A*100+0.5)/100
480 PRINT A
490 Q2$="COMPUTE":GOSUB 3400
500 IF YN$="N" THEN RETURN
510 GOTO 290
```

Figure 2.4: Program Listing: Future Value of a Deposit

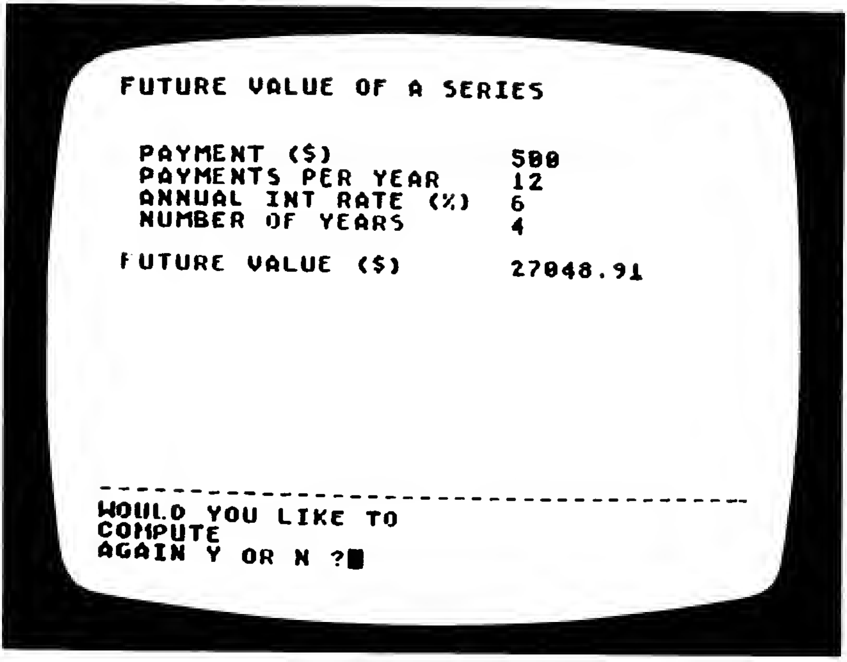
FUTURE VALUE OF A SERIES OF DEPOSITS

Description

This program calculates the future value of a series of regular deposits. For example, you can determine the future value of an IRA if you make regular annual contributions. You specify the amount of the payment, the number of payments made per year, the annual interest rate your funds earn, and the number of years you will make the payments. The program assumes that the compounding period and number of payments per year are the same, and that the annual interest rate is constant.

Example

Ann plans to deposit $500 in a savings account every month. Interest is earned at 6% per annum and is compounded monthly. How much money will be in her account in four years? Figure 2.5 shows the screen display for this example. The program listing is in Figure 2.6.



```
FUTURE VALUE OF A SERIES

 PAYMENT ($)               500
 PAYMENTS PER YEAR         12
 ANNUAL INT RATE (%)       6
 NUMBER OF YEARS           4

FUTURE VALUE ($)           27048.91




---------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN Y OR N ?
```

Figure 2.5: Screen Display: Future Value of a Series of Deposits

```
520 REM ------------------------------
530 N$="FUTURE VALUE OF A SERIES"
540 REM ------------------------------
550 GRAPHICS 0:PRINT N$:PRINT :K=1
560 REM ASK FOR PARAMETERS
570 Q1$="ENTER DATA":Q2$=""
580 Q3$="PAYMENT ($)":GOSUB 6600
590 Q3$="PAYMENTS PER YEAR":GOSUB 6600
600 Q3$="ANNUAL INT RATE (%)":GOSUB 6600
610 Q3$="NUMBER OF YEARS":GOSUB 6600
620 REM COMPUTE RESULT
630 PRINT :PRINT "FUTURE VALUE ($)";:POSITION 24,8
640 A=0
650 FOR I=1 TO P(2)*P(4)
660 A=P(1)+A+A*P(3)/100/P(2)
670 NEXT I
680 A=INT(A*100+0.5)/100
690 PRINT A
700 Q2$="COMPUTE":GOSUB 3400:REM ASK FOR ANOTHER
710 IF YN$="N" THEN RETURN
720 GOTO 520
```

Figure 2.6: Program Listing: Future Value of a Series of Deposits

PRESENT VALUE OF AN AMOUNT

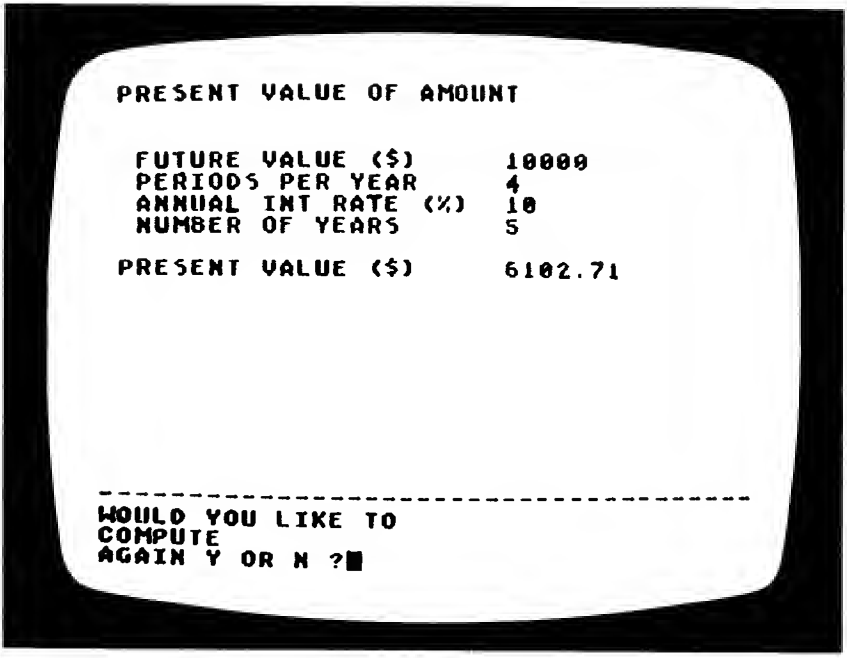
Description

Assume that Bob owes you $5,000 in three years. He offers to pay you $4,000 today, and you want to know whether to accept his offer. The Present Value program uses an interest rate you specify to find the present value of a future amount. You should accept Bob's offer if the present value is less than $4,000.

To use this program you supply the future value, the annual interest rate at which your funds earn money, the number of compounding periods per year, and the number of years in the future that the value of the amount is known.

Example

Bill has a note for which he will receive $10,000 five years from now. Bill normally earns 10% interest on his funds, with interest being compounded quarterly. Bill would like to know the present value of his note. Figure 2.7 shows the screen display for the Present Value program; Figure 2.8 lists the Present Value program.



```
PRESENT VALUE OF AMOUNT

 FUTURE VALUE ($)      10000
 PERIODS PER YEAR      4
 ANNUAL INT RATE (%)   10
 NUMBER OF YEARS       5

PRESENT VALUE ($)      6102.71




----------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN Y OR N ?█
```

Figure 2.7: Screen Display: Present Value of an Amount

```
730 REM ------------------------------
740 N$="PRESENT VALUE OF AMOUNT"
750 REM ------------------------------
760 GRAPHICS 0:PRINT N$:PRINT :K=1:REM INITIALIZE
770 REM ASK FOR PARAMETERS
780 Q1$="ENTER DATA":Q2$=""
790 Q3$="FUTURE VALUE ($)":GOSUB 6600
800 Q3$="PERIODS PER YEAR":GOSUB 6600
810 Q3$="ANNUAL INT RATE (%)":GOSUB 6600
820 Q3$="NUMBER OF YEARS":GOSUB 6600
830 REM DO THE COMPUTATION
840 PRINT :PRINT "PRESENT VALUE ($)":POSITION 24,8
850 A=P(1)/(1+P(3)/100/P(2))^(P(2)*P(4))
860 A=INT(A*100+0.5)/100
870 PRINT A
880 Q2$="COMPUTE":GOSUB 3400:REM ASK FOR ANOTHER
890 IF YN$="N" THEN RETURN
900 GOTO 730
```

Figure 2.8: Program Listing: Present Value of an Amount

PRESENT VALUE OF A SERIES OF PAYMENTS

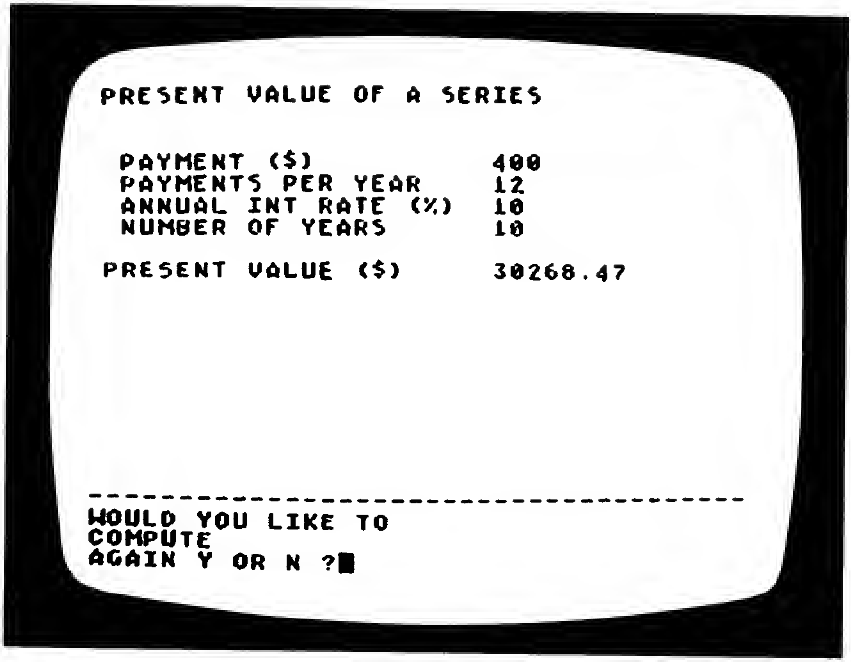
Description

The previous example shows you how to find the present value of a future amount. With this program you can find the present value of a series of payments. For example, assume that Sally buys your car and pays you $50 a month for three years. After one year she offers you $900 in cash, and you want to know whether to accept her offer. You can use this program to find the answer to your question. You should accept if the present value is less than $900.

To use this program you specify the amount of the payment, the number of payments you receive per year, the number of years for which the payments will continue, and the annual interest rate at which your funds earn interest. We assume that the number of payments per year and the number of compounding periods are identical.

Example

Jackie receives $400 per month in payments. She can earn 10% interest on invested funds, and she wants to know the present value of the payments. The payments will continue for ten years. Figure 2.9 shows the screen display for this example, and Figure 2.10 is the complete program listing.



```
PRESENT VALUE OF A SERIES

 PAYMENT ($)             400
 PAYMENTS PER YEAR       12
 ANNUAL INT RATE (%)     10
 NUMBER OF YEARS         10

PRESENT VALUE ($)        30268.47








---------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN Y OR N ?█
```

Figure 2.9: Screen Display: Present Value of a Series of Payments

```
910 REM ------------------------------
920 N$="PRESENT VALUE OF A SERIES"
930 REM ------------------------------
940 GRAPHICS 0:PRINT N$:PRINT :K=1:REM INITIALIZE
950 REM ASK FOR PARAMETERS
955 Q1$="ENTER DATA":Q2$=""
960 Q3$="PAYMENT ($)":GOSUB 6600
970 Q3$="PAYMENTS PER YEAR":GOSUB 6600
980 Q3$="ANNUAL INT RATE (%)":GOSUB 6600
990 Q3$="NUMBER OF YEARS":GOSUB 6600
1000 REM DO THE COMPUTATION
1010 PRINT :PRINT "PRESENT VALUE ($)":POSITION 24,8
1020 A=0
1030 FOR I=1 TO P(2)*P(4)
1040 A=A+P(1)/((1+P(3)/100/P(2))^I)
1050 NEXT I
1060 A=INT(A*100+0.5)/100
1070 PRINT A
1080 Q2$="COMPUTE":GOSUB 3400:REM ASK FOR ANOTHER
1090 IF YN$="N" THEN RETURN
1100 GOTO 910
```

Figure 2.10: Program Listing: Present Value of a Series of Payments

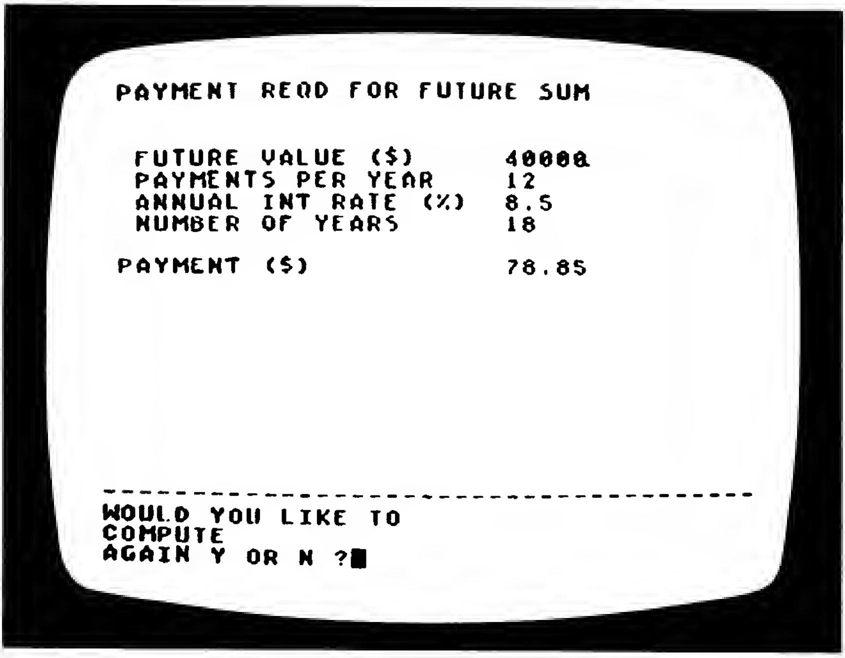
PAYMENT REQUIRED FOR FUTURE SUM

Description

This program allows you to determine the regular payment necessary to produce a required amount of money in the future. For example, you can use this program to calculate what regular payments will be necessary to take a special vacation, replace a piece of equipment, or save for a college education. You must specify the future sum, the number of payments per year, the annual interest rate at which your funds earn interest, and the number of years to the time you need the money. The program assumes that the number of payments and the compounding periods are identical.

Example

Bob plans to make monthly payments into a college fund for his daughter. He will make the payments for 18 years, and he estimates that he will need $40,000. He can earn 8.5% interest on his college savings account. We show the screen display for this example in Figure 2.11 and the program listing in Figure 2.12.



```
PAYMENT REQD FOR FUTURE SUM

 FUTURE VALUE ($)       40000
 PAYMENTS PER YEAR      12
 ANNUAL INT RATE (%)    8.5
 NUMBER OF YEARS        18

PAYMENT ($)             78.85




--------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN Y OR N ?█
```

Figure 2.11: Screen Display: Payment Required for Future Sum

```
1110 REM ----------------------------
1120 N$="PAYMENT REQD FOR FUTURE SUM"
1130 REM ----------------------------
1140 GRAPHICS 0:PRINT N$:PRINT :K=1
1150 REM ASK FOR PARAMETERS
1160 Q1$="ENTER DATA":Q2$=""
1170 Q3$="FUTURE VALUE ($)":GOSUB 6600
1180 Q3$="PAYMENTS PER YEAR":GOSUB 6600
1190 Q3$="ANNUAL INT RATE (%)":GOSUB 6600
1200 Q3$="NUMBER OF YEARS":GOSUB 6600
1210 REM DO THE COMPUTATION
1220 PRINT :PRINT "PAYMENT ($)":POSITION 24,8
1230 I=P(3)/100/P(2)
1240 Q=P(2)*P(4)
1250 A=P(1)*I/((1+I)^Q-1)
1260 A=INT(A*100+0.5)/100
1270 PRINT A
1280 Q2$="COMPUTE":GOSUB 3400:REM ASK FOR ANOTHER
1290 IF YN$="N" THEN RETURN
1300 GOTO 1110
```

Figure 2.12: Program Listing: Payment Required for Future Sum

WITHDRAWAL OF FUNDS

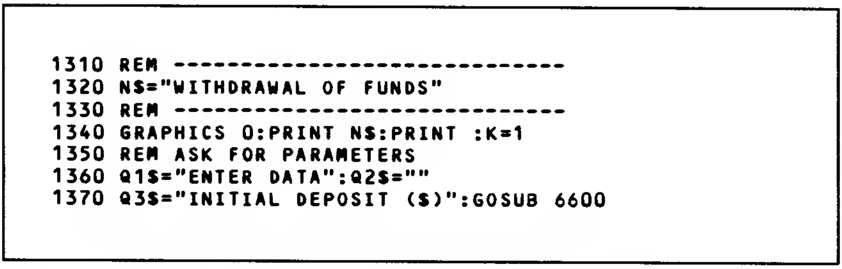
Description

You can use this program to determine how long you can make withdrawals from a bank account before the account runs out of funds. You must specify the initial deposit, the number of withdrawals per year, the interest rate at which the remaining funds earn interest, and the amount of the withdrawal. Again, the program assumes that the number of withdrawals per year and the number of compounding periods are identical.

If the withdrawal amount is less than the interest earned, your bank account will grow rather than run out of funds. The program then displays the message BANK BALANCE IS GROWING. Don't be concerned if this program takes a long time to finish running; the program executes as many times as necessary to find a zero bank balance.

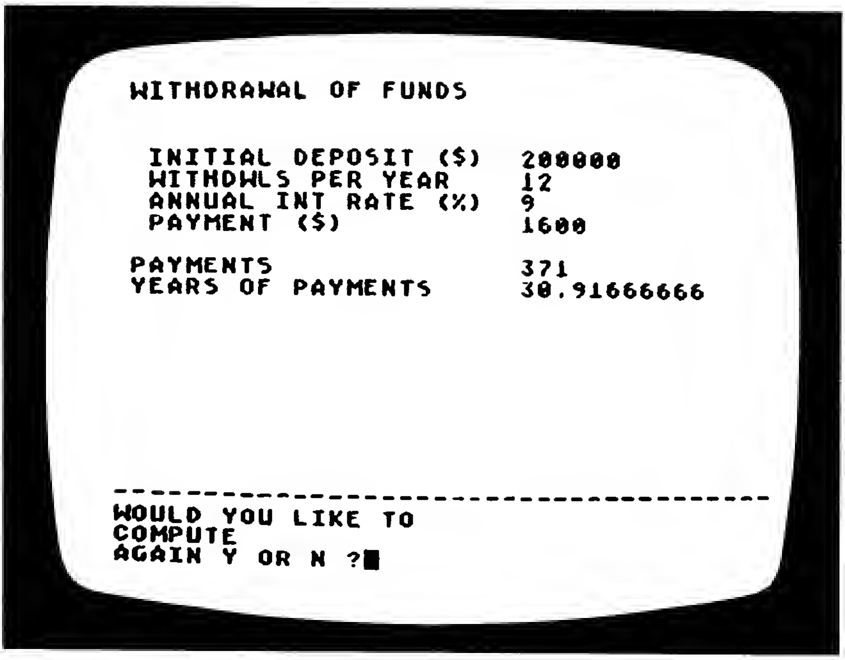
Example

Jack and Jill Jones are ready to retire. They have $200,000 in their bank account, and the account earns 9% interest. Jack and Jill plan to live on $1,600 per month. How long will they be able to withdraw money? Figure 2.13 shows the screen display for this example; the program is listed in Figure 2.14.



```
1310 REM -----------------------------
1320 N$="WITHDRAWAL OF FUNDS"
1330 REM -----------------------------
1340 GRAPHICS 0:PRINT N$:PRINT :K=1
1350 REM ASK FOR PARAMETERS
1360 Q1$="ENTER DATA":Q2$=""
1370 Q3$="INITIAL DEPOSIT ($)":GOSUB 6600
```

Figure 2.14: Program Listing: Withdrawal of Funds (continues)



```
WITHDRAWAL OF FUNDS

 INITIAL DEPOSIT ($)   200000
 WITHDWLS PER YEAR     12
 ANNUAL INT RATE (%)   9
 PAYMENT ($)           1600

PAYMENTS               371
YEARS OF PAYMENTS      30.91666666







---------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN Y OR N ?█
```

Figure 2.13: Screen Display: Withdrawal of Funds

```
1380 Q3$="WITHDWLS PER YEAR":GOSUB 6600
1390 Q3$="ANNUAL INT RATE (%)":GOSUB 6600
1400 Q3$="PAYMENT ($)":GOSUB 6600
1410 REM DO THE COMPUTATION
1420 PRINT :PRINT "PAYMENTS":POSITION 24,8
1430 A=P(1):N=1
1440 A=A+A*P(3)/100/P(2)-P(4)
1450 IF A<P(1) THEN 1500
1460 PRINT :PRINT :PRINT "BANK BALANCE IS GROWING"
1470 FOR X=100 TO 1 STEP -3
1475 SOUND 1,X,10,15:SOUND 2,X+50,10,15
1480 SOUND 3,X+100,10,15:NEXT X
1485 SOUND 1,0,0,0:SOUND 2,0,0,0:SOUND 3,0,0,0
1490 GOTO 1550
1500 IF A<P(4) THEN 1530
1510 N=N+1
1520 GOTO 1440
1530 PRINT N:PRINT "YEARS OF PAYMENTS"
1540 POSITION 24,9:PRINT N/12
1550 Q2$="COMPUTE":GOSUB 3400
1560 IF YN$="N" THEN RETURN
1570 GOTO 1310
```

Figure 2.14: Program Listing: Withdrawal of Funds

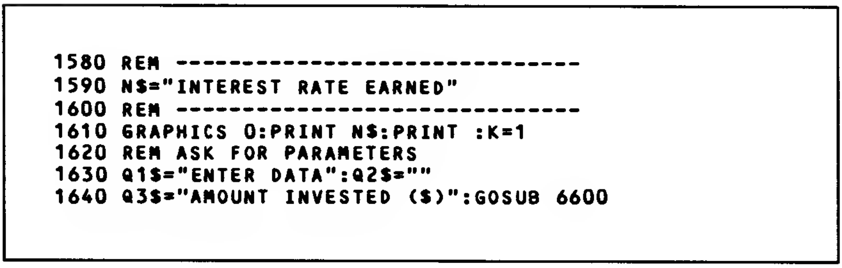
INTEREST RATE EARNED

Description

The program in this example is useful when you want to determine the interest rate you earn when you know the beginning and ending values of your investment. For example, you can find the equivalent interest rate on the profit from a house sale. To use this program you specify the beginning and ending values of your funds, the number of compounding periods per year, and the number of years over which your funds have grown.

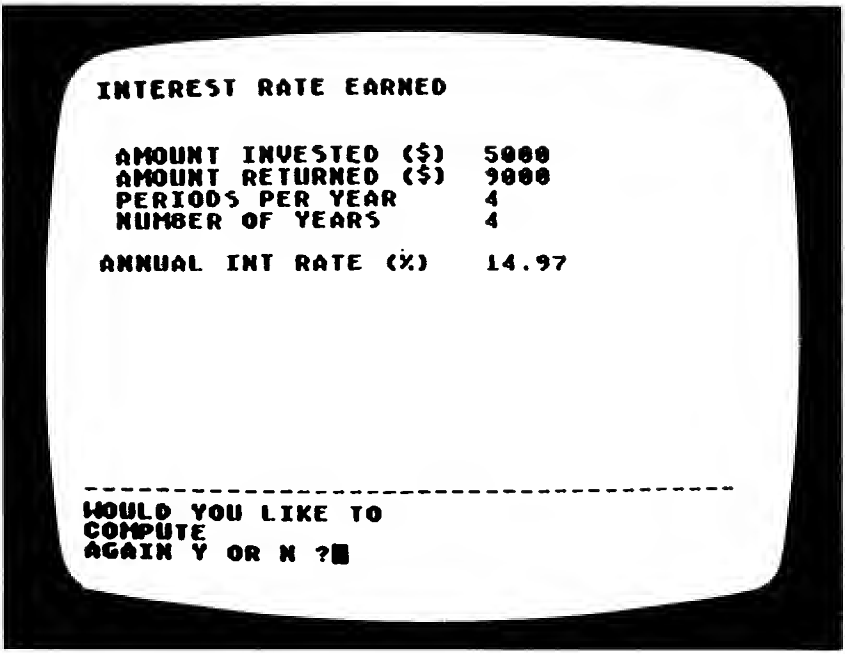
Example

Auntie Mame gave $5,000 to her broker to invest four years ago. She has just received $9,000 in return. She wishes to compute the equivalent interest rate she has earned, with interest compounded quarterly. Figures 2.15 and 2.16 show the screen display and program listing for this example.



```
1580 REM -----------------------------
1590 N$="INTEREST RATE EARNED"
1600 REM -----------------------------
1610 GRAPHICS 0:PRINT N$:PRINT :K=1
1620 REM ASK FOR PARAMETERS
1630 Q1$="ENTER DATA":Q2$=""
1640 Q3$="AMOUNT INVESTED ($)":GOSUB 6600
```

Figure 2.16: Program Listing: Interest Rate Earned (continues)



```
INTEREST RATE EARNED

 AMOUNT INVESTED ($)   5000
 AMOUNT RETURNED ($)   9000
 PERIODS PER YEAR      4
 NUMBER OF YEARS       4

ANNUAL INT RATE (%)    14.97





--------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN Y OR N ?█
```

Figure 2.15: Screen Display: Interest Rate Earned

```
1650 Q3$="AMOUNT RETURNED ($)":GOSUB 6600
1660 Q3$="PERIODS PER YEAR":GOSUB 6600
1670 Q3$="NUMBER OF YEARS":GOSUB 6600
1680 REM DO THE COMPUTATION
1690 PRINT :PRINT "ANNUAL INT RATE (%)"
1700 POSITION 24,8:Q=P(3)*P(4)
1710 I=((P(2)/P(1))^(1/Q)-1)*100*P(3)
1720 I=INT(I*100+0.5)/100
1730 PRINT I
1740 Q2$="COMPUTE":GOSUB 3400
1750 IF YN$="N" THEN RETURN
1760 GOTO 1580
```

Figure 2.16: Program Listing: Interest Rate Earned

NET PRESENT VALUE, UNEVEN CASH FLOWS

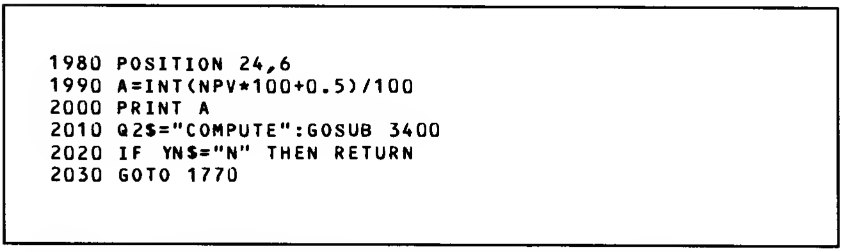
Description

Net present value (NPV) analysis is used by financial analysts to determine whether an investment is worthwhile. For an interest rate you specify, a positive NPV indicates that the investment is desirable.

This program is useful when you wish to determine the present value of a series of cash flows. The cash flows need not be the same in every year. You use this program by entering the initial investment you make and the annual interest rate at which your funds earn interest. The program then prompts you for each cash flow. You enter these cash flows as positive or negative numbers, depending on whether you receive or distribute funds. You signal the end of cash flow entries by typing 0.

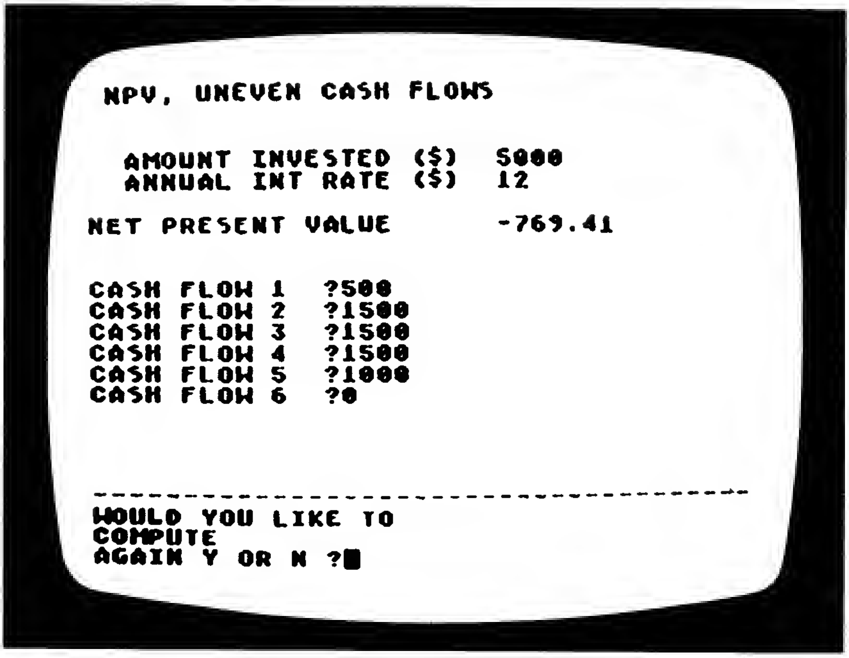
Example

Tina can buy a note for $5,000. She will receive annual payments of $500, $1,500, $1,500, $1,500, and $1,000. Tina would like her funds to earn 12% annual interest. What is the net present value of her investment? The display in Figure 2.17 is negative, indicating that she is not earning 12%; thus she should not buy the note. A complete program listing is shown in Figure 2.18.



```
1980 POSITION 24,6
1990 A=INT(NPV*100+0.5)/100
2000 PRINT A
2010 Q2$="COMPUTE":GOSUB 3400
2020 IF YN$="N" THEN RETURN
2030 GOTO 1770
```

Figure 2.18: Program Listing: Net Present Value, Uneven Cash Flows (continues)



```
NPV, UNEVEN CASH FLOWS

  AMOUNT INVESTED ($)   5000
  ANNUAL INT RATE ($)   12

NET PRESENT VALUE        -769.41


CASH FLOW 1   ?500
CASH FLOW 2   ?1500
CASH FLOW 3   ?1500
CASH FLOW 4   ?1500
CASH FLOW 5   ?1000
CASH FLOW 6   ?0



---------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN Y OR N ?█
```

Figure 2.17: Screen Display: Net Present Value, Uneven Cash Flows

```
1770 REM ------------------------------
1780 N$="NPV, UNEVEN CASH FLOWS"
1790 REM ------------------------------
1800 REM INITIALIZE
1810 GRAPHICS 0:PRINT N$:PRINT :K=1
1820 REM ASK FOR PARAMETERS
1830 Q1$="ENTER DATA":Q2$=""
1840 Q3$="AMOUNT INVESTED ($)":GOSUB 6600
1850 Q3$="ANNUAL INT RATE ($)":GOSUB 6600
1860 Q1$="ENTER ANNUAL CASH FLOWS"
1870 Q2$="IN ORDER RECIEVED"
1880 Q3$="TYPE ZERO WHEN DONE"
1890 GOSUB 5000:REM ASK QUESTIONS
1900 NPV=-P(1):FOR I=1 TO 100
1910 POSITION 1,I-INT(I/10)*10+8
1920 PRINT "CASH FLOW ";I;"          ";
1930 POSITION 14,I-INT(I/10)*10+8
1940 INPUT X:REM STORE DATA
1950 IF X=0 THEN 1970
1955 NPV=NPV+X/(1+P(2)/100)^I
1960 NEXT I
1970 POSITION 1,6:PRINT "NET PRESENT VALUE"
```

Figure 2.18: Program Listing: Net Present Value, Uneven Cash Flows

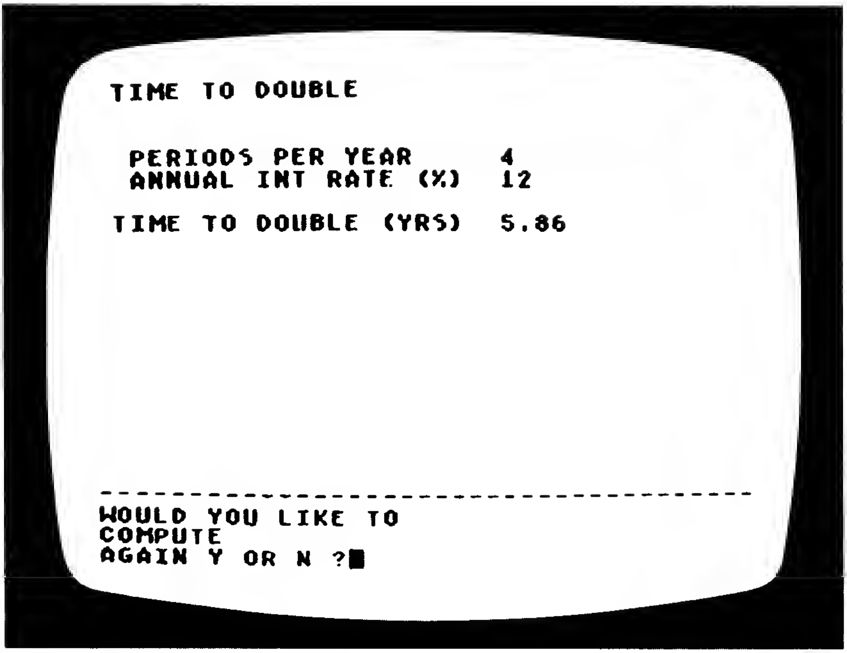
TIME TO DOUBLE

Description

This program computes the time it will take for the funds you invest to double. To use the program you enter the annual interest rate and number of compounding periods per year.

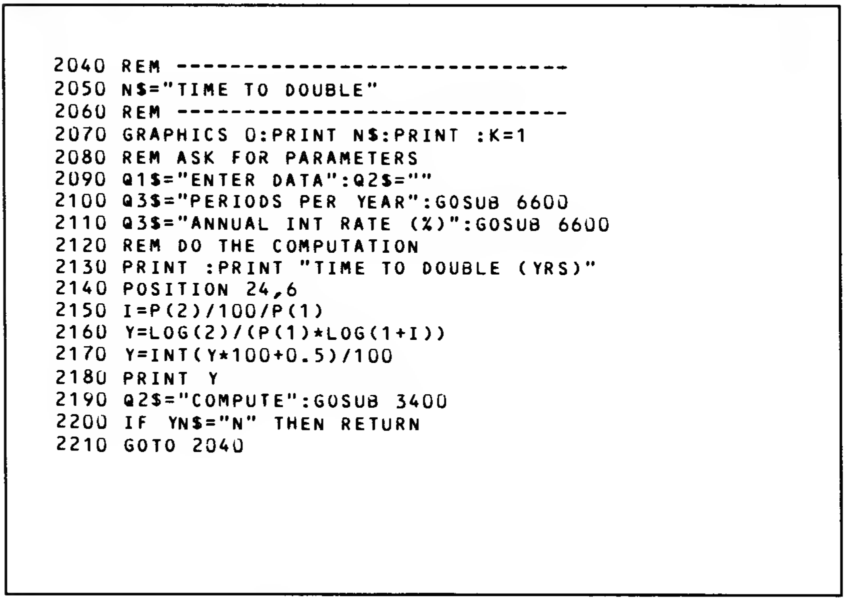
Example

Fred's bank compounds interest four times yearly and pays 12% interest per annum. How many years will it take Fred to double a $5,000 investment? We display the results of this program in Figure 2.19; a program listing is shown in Figure 2.20.



```
TIME TO DOUBLE

 PERIODS PER YEAR      4
 ANNUAL INT RATE (%)   12

TIME TO DOUBLE (YRS)   5.86








--------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN Y OR N ?█
```

Figure 2.19: Screen Display: Time to Double



```
2040 REM ----------------------------
2050 N$="TIME TO DOUBLE"
2060 REM ----------------------------
2070 GRAPHICS 0:PRINT N$:PRINT :K=1
2080 REM ASK FOR PARAMETERS
2090 Q1$="ENTER DATA":Q2$=""
2100 Q3$="PERIODS PER YEAR":GOSUB 6600
2110 Q3$="ANNUAL INT RATE (%)":GOSUB 6600
2120 REM DO THE COMPUTATION
2130 PRINT :PRINT "TIME TO DOUBLE (YRS)"
2140 POSITION 24,6
2150 I=P(2)/100/P(1)
2160 Y=LOG(2)/(P(1)*LOG(1+I))
2170 Y=INT(Y*100+0.5)/100
2180 PRINT Y
2190 Q2$="COMPUTE":GOSUB 3400
2200 IF YN$="N" THEN RETURN
2210 GOTO 2040
```

Figure 2.20: Program Listing: Time to Double

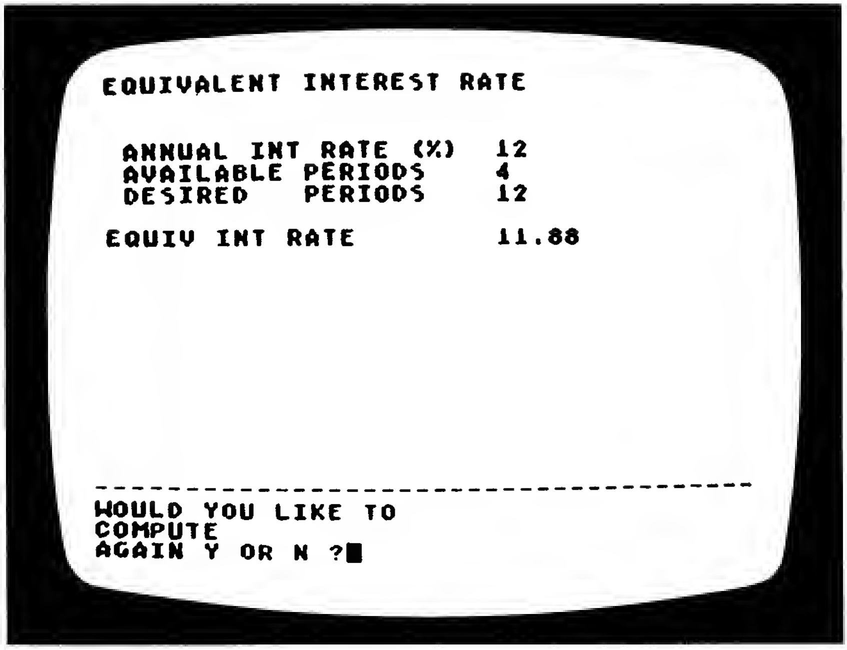
EQUIVALENT INTEREST RATE

Description

Many of the programs in this chapter assume that the number of compounding periods and the number of payments are identical. This program allows you to determine an equivalent interest rate when the two periods are not the same. To use the program, enter the annual interest rate, the number of compounding periods, and the number of payments you want to make per year (desired periods). The program then computes an equivalent interest rate that you can use.

Example

You make deposits monthly into an account, and you wish to determine the future value of the deposits. The interest on this account is 12% per annum, compounded quarterly. As shown in Figure 2.21, the equivalent interest rate is 11.88%. You should use this figure and then run the Future Value of a Series of Deposits program to compute your final answer. The program listing for this example is given in Figure 2.22.



```
EQUIVALENT INTEREST RATE

 ANNUAL INT RATE (%)   12
 AVAILABLE PERIODS     4
 DESIRED   PERIODS     12

EQUIV INT RATE         11.88




---------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN Y OR N ?█
```

Figure 2.21: Screen Display: Equivalent Interest Rate

```
2320 REM ----------------------------
2330 N$="EQUIVALENT INTEREST RATE"
2340 REM ----------------------------
2350 GRAPHICS 0:PRINT N$:PRINT :K=1
2360 REM ASK FOR PARAMETERS
2370 Q1$="ENTER DATA":Q2$=""
2380 Q3$="ANNUAL INT RATE (%)":GOSUB 6600
2390 Q3$="AVAILABLE PERIODS":GOSUB 6600
2400 Q3$="DESIRED   PERIODS":GOSUB 6600
2410 I=P(1):Q2=P(2):Q1=P(3)
2420 IEQ=Q1*(1+I/100/Q2)^(Q2/Q1)-Q1
2430 PRINT :PRINT "EQUIV INT RATE"
2440 POSITION 24,7
2450 IEQ=INT(10000*IEQ+0.5)/100
2460 PRINT IEQ
2470 Q2$="COMPUTE":GOSUB 3400
2480 IF YN$="N" THEN RETURN
2490 GOTO 2320
```

Figure 2.22: Program Listing: Equivalent Interest Rate

3
USEFUL BUSINESS PROGRAMS

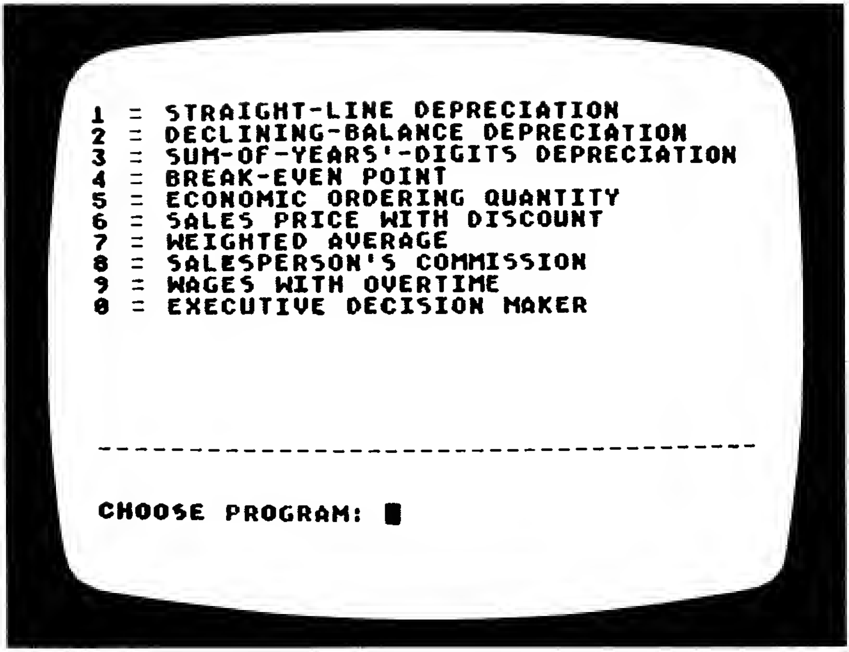
This chapter presents ten helpful business programs. These programs enable you to prepare depreciation schedules, find the economic ordering quantity, and solve many other business problems.

For example, you may be starting a business or running a charity event; the Break-Even Point program will calculate how many items or tickets you must sell to produce a profit. Whether you are a salesperson or employ a salesperson, the Salesperson's Commission program allows you to find the commission due for a given sales volume. The Wages with Overtime program makes it simple to calculate the salary due when overtime is worked. This program adds base pay and overtime pay to find the total salary due. We also provide an executive decision making program, which can be used when all else fails.

Remember to save these programs as you enter them and to merge the central subroutines prior to running the programs.

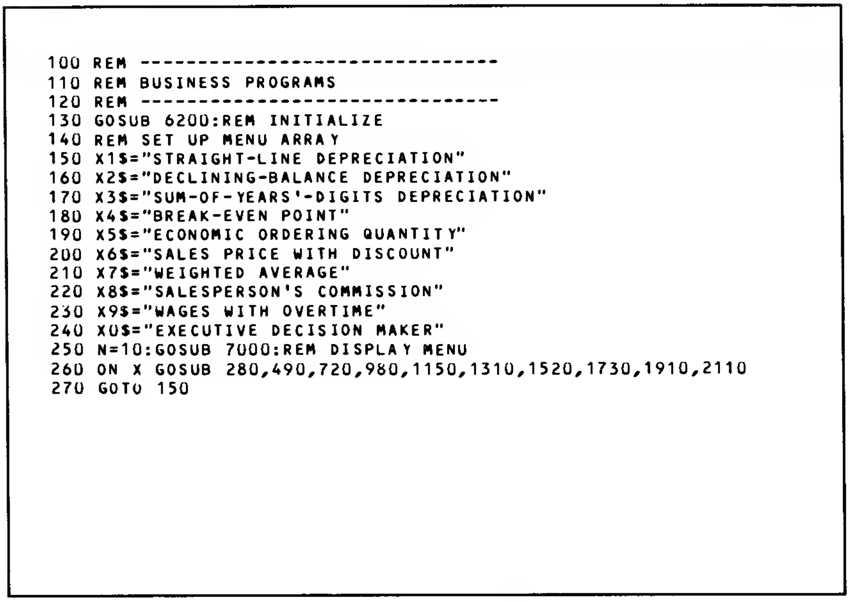
BUSINESS PROGRAMS MENU

Before you enter the business programs, you must build the menu. You then select and run your programs from the menu. Review the menu entry instructions in Chapter 1 prior to creating your menu. Figures 3.1 and 3.2 display an image of the menu and the menu selection program.



```
1 = STRAIGHT-LINE DEPRECIATION
2 = DECLINING-BALANCE DEPRECIATION
3 = SUM-OF-YEARS'-DIGITS DEPRECIATION
4 = BREAK-EVEN POINT
5 = ECONOMIC ORDERING QUANTITY
6 = SALES PRICE WITH DISCOUNT
7 = WEIGHTED AVERAGE
8 = SALESPERSON'S COMMISSION
9 = WAGES WITH OVERTIME
0 = EXECUTIVE DECISION MAKER



----------------------------------------

CHOOSE PROGRAM: █
```

Figure 3.1: Screen Display: Business Programs Menu



```
100 REM -------------------------------
110 REM BUSINESS PROGRAMS
120 REM -------------------------------
130 GOSUB 6200:REM INITIALIZE
140 REM SET UP MENU ARRAY
150 X1$="STRAIGHT-LINE DEPRECIATION"
160 X2$="DECLINING-BALANCE DEPRECIATION"
170 X3$="SUM-OF-YEARS'-DIGITS DEPRECIATION"
180 X4$="BREAK-EVEN POINT"
190 X5$="ECONOMIC ORDERING QUANTITY"
200 X6$="SALES PRICE WITH DISCOUNT"
210 X7$="WEIGHTED AVERAGE"
220 X8$="SALESPERSON'S COMMISSION"
230 X9$="WAGES WITH OVERTIME"
240 X0$="EXECUTIVE DECISION MAKER"
250 N=10:GOSUB 7000:REM DISPLAY MENU
260 ON X GOSUB 280,490,720,980,1150,1310,1520,1730,1910,2110
270 GOTO 150
```

Figure 3.2: Program Listing: Business Programs Menu

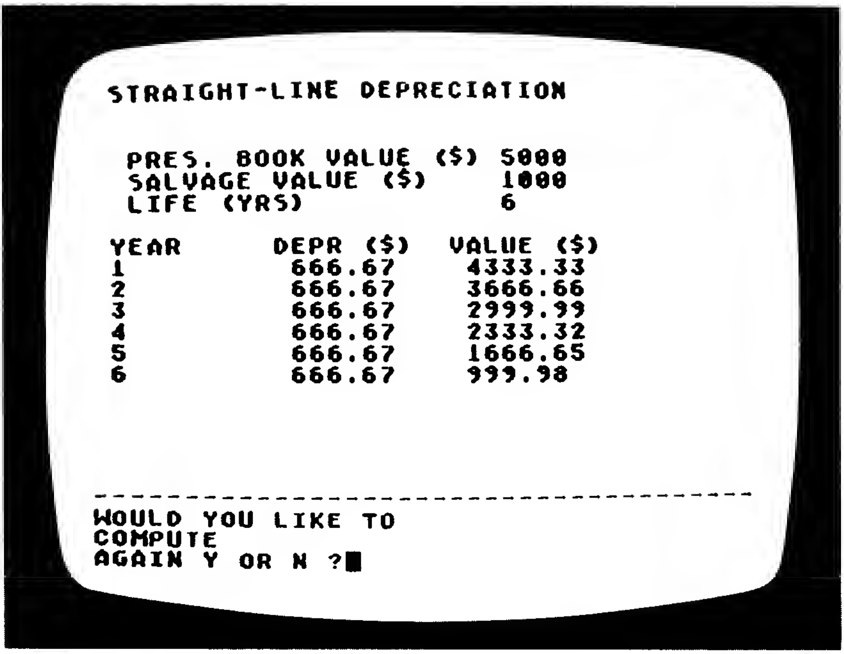
STRAIGHT-LINE DEPRECIATION

Description

This program computes a yearly depreciation schedule based on the straight-line method. To depreciate an asset, you specify the present book value, the estimated salvage value, and the life of the asset in years. If your asset has a life longer than ten years, the program displays the schedule for the first ten years and then pauses. When you are ready to review years 11–20, you simply press any key, and the schedule for those years is displayed.

Example

Let's set up a depreciation schedule for an office copier. We presently value the copier at $5,000. We expect to use the copier for six years and then sell it for $1,000. Figure 3.3 shows the screen display of the depreciation schedule. Figure 3.4 is a listing of the depreciation program.



```
STRAIGHT-LINE DEPRECIATION

 PRES. BOOK VALUE ($) 5000
 SALVAGE VALUE ($)    1000
 LIFE (YRS)           6

YEAR      DEPR ($)  VALUE ($)
1          666.67    4333.33
2          666.67    3666.66
3          666.67    2999.99
4          666.67    2333.32
5          666.67    1666.65
6          666.67    999.98




--------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN Y OR N ?█
```

Figure 3.3: Screen Display: Straight-Line Depreciation

```
280 REM ----------------------------
290 N$="STRAIGHT-LINE DEPRECIATION"
300 REM ----------------------------
310 GRAPHICS 0:PRINT N$:PRINT :K=1
320 REM GET INPUT PARAMETERS
330 Q1$="ENTER DATA":Q2$=""
340 Q3$="PRES. BOOK VALUE ($)":GOSUB 6600
350 Q3$="SALVAGE VALUE ($)":GOSUB 6600
360 Q3$="LIFE (YRS)":GOSUB 6600
370 REM PRINT DEPR SCHEDULE
380 D=(P(1)-P(2))/P(3)
390 PRINT
400 PRINT "YEAR     DEPR ($)  VALUE ($)"
410 FOR Y=1 TO P(3)
413 L=Y+7-INT((Y-1)/10)*10
415 POSITION 1,L:PRINT S$:POSITION 2,L
420 PRINT Y,
430 D=INT(D*100+0.5)/100:PRINT D,P(1)-D*Y
440 IF Y/10=INT(Y/10) THEN GOSUB 7400
450 NEXT Y
460 Q2$="COMPUTE":GOSUB 3400:REM ASK FOR ANOTHER
470 IF YN$="N" THEN RETURN
480 GOTO 310
```

Figure 3.4: Program Listing: Straight-Line Depreciation

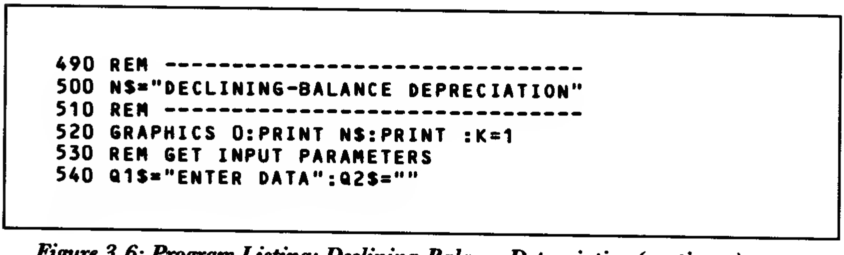
DECLINING-BALANCE DEPRECIATION

Description

This program makes it easy to calculate depreciation schedules based on the declining-balance method, which is an accelerated method of depreciation. Accelerated depreciation methods are often used to provide more favorable tax treatment than the straight-line method provides. The declining-balance method multiplies the remaining book value of an asset by an acceleration factor and then divides by the life of the asset in years. You must specify the initial book value of the asset, its useful life, and the acceleration factor. Typical acceleration factors are 150%, 175%, and 200%; the declining-balance method ignores the salvage value.

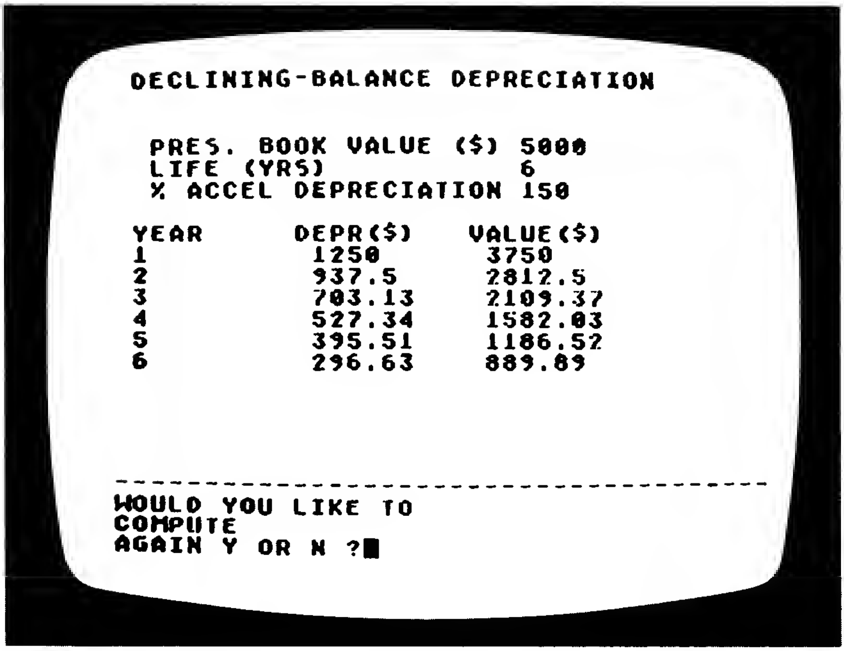
Example

Let's now depreciate the office copier in the previous example by the declining-balance method. In this example we use an acceleration factor of 150%. The depreciation schedule appears in Figure 3.5 and the program listing in Figure 3.6.



```
490 REM ------------------------------
500 N$="DECLINING-BALANCE DEPRECIATION"
510 REM ------------------------------
520 GRAPHICS 0:PRINT N$:PRINT :K=1
530 REM GET INPUT PARAMETERS
540 Q1$="ENTER DATA":Q2$=""
```

Figure 3.6: Program Listing: Declining-Balance Depreciation (continues)



```
DECLINING-BALANCE DEPRECIATION

 PRES. BOOK VALUE ($) 5000
 LIFE (YRS)           6
 % ACCEL DEPRECIATION 150

YEAR      DEPR($)    VALUE($)
1          1250       3750
2          937.5      2812.5
3          703.13     2109.37
4          527.34     1582.03
5          395.51     1186.52
6          296.63     889.89




-------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN Y OR N ?█
```

Figure 3.5: Screen Display: Declining-Balance Depreciation

```
550 Q3$="PRES. BOOK VALUE ($)":GOSUB 6600
560 Q3$="LIFE (YRS)":GOSUB 6600
570 Q3$="% ACCEL DEPRECIATION":GOSUB 6600
580 REM PRINT DEPR SCHEDULE
590 RV=P(1)
600 PRINT
610 PRINT "YEAR     DEPR($)   VALUE($)"
620 FOR Y=1 TO P(2)
625 L=Y+7-INT((Y-1)/10)*10
626 POSITION 1,L:PRINT S$:POSITION 2,L
630 PRINT Y,
640 D=RV*P(3)/100/P(2)
650 D=INT(D*100+0.5)/100:PRINT D,RV-D
660 RV=RV-D
665 REM PAUSE AT SCREENFUL?
670 IF Y/10=INT(Y/10) THEN GOSUB 7400
680 NEXT Y
690 Q2$="COMPUTE":GOSUB 3400
700 IF YN$="N" THEN RETURN
710 GOTO 490
```

Figure 3.6: Program Listing: Declining-Balance Depreciation

SUM-OF-YEARS'-DIGITS DEPRECIATION

Description

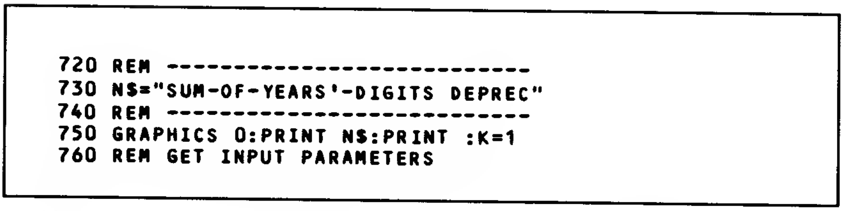
We now show a second method of accelerated depreciation, the sum of the years' digits. This depreciation method multiplies the remaining book value by the ratio of the number of remaining years to the sum of the years of life. If the life of the asset is five years, the sum of the years of life is

$$5 + 4 + 3 + 2 + 1 = 15$$

You again specify the present book value, salvage value, and life of the asset. Your computer simplifies this otherwise tedious calculation.

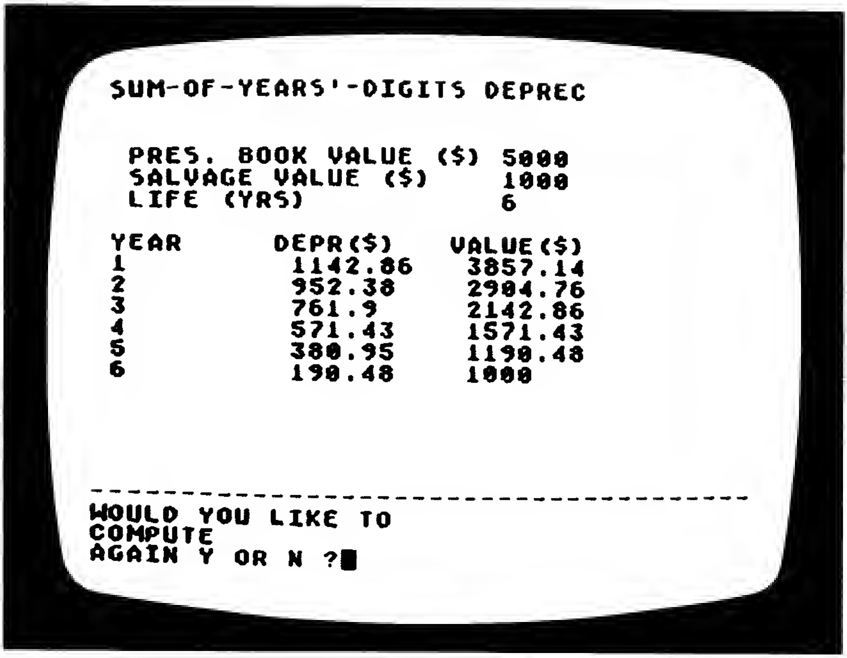
Example

Let us produce a sum-of-the-years'-digits depreciation schedule for the office copier in the prior examples. The schedule is shown in Figure 3.7; the program listing appears in Figure 3.8.



```
720 REM -------------------------
730 N$="SUM-OF-YEARS'-DIGITS DEPREC"
740 REM -------------------------
750 GRAPHICS 0:PRINT N$:PRINT :K=1
760 REM GET INPUT PARAMETERS
```

Figure 3.8: Program Listing: Sum-of-Years'-Digits Depreciation (continues)



```
SUM-OF-YEARS'-DIGITS DEPREC

 PRES. BOOK VALUE ($) 5000
 SALVAGE VALUE ($)    1000
 LIFE (YRS)           6

YEAR      DEPR($)    VALUE($)
1          1142.86    3857.14
2          952.38     2904.76
3          761.9      2142.86
4          571.43     1571.43
5          380.95     1190.48
6          190.48     1000



----------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN Y OR N ?█
```

Figure 3.7: Screen Display: Sum-of-Years'-Digits Depreciation

```
770 Q1$="ENTER DATA":Q2$=""
780 Q3$="PRES. BOOK VALUE ($)":GOSUB 6600
790 Q3$="SALVAGE VALUE ($)":GOSUB 6600
800 Q3$="LIFE (YRS)":GOSUB 6600
810 REM PRINT DEPR SCHEDULE
820 D=P(1)-P(2)
830 RV=P(1)
840 PRINT
850 PRINT "YEAR     DEPR($)   VALUE($)"
860 FOR Y=1 TO P(3)
870 L=Y+7-INT((Y-1)/10)*10
880 POSITION 1,L:PRINT S$:POSITION 2,L
890 D1=D*(P(3)+1-Y)/(P(3)*(P(3)+1)/2)
900 D1=INT(D1*100+0.5)/100
910 PRINT Y,D1,RV-D1
920 RV=RV-D1
925 REM PAUSE AT SCREENFUL?
930 IF Y/10=INT(Y/10) THEN GOSUB 7400
940 NEXT Y
950 Q2$="COMPUTE":GOSUB 3400:REM ASK FOR ANOTHER
960 IF YN$="N" THEN RETURN
970 GOTO 720
```

Figure 3.8: Program Listing: Sum-of-Years'-Digits Depreciation

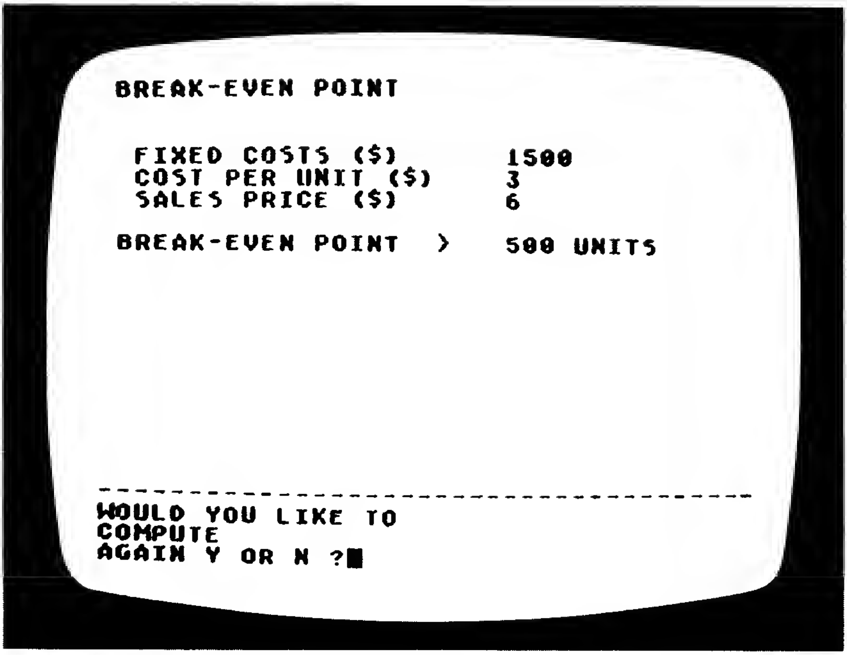
BREAK-EVEN POINT

Description

This program allows you to find the number of items you must sell for a business or product line to break even. To use this program you must specify the fixed costs of the business (overhead), the cost per unit to manufacture or buy your product, and the sales price per unit.

Example

Harry buys novelty items for $3.00 each and plans to sell them for $6.00 each through mail order. If Harry plans to spend $1,500 to promote and sell the items, how many units must he sell to break even? The screen display for this example is shown in Figure 3.9; the program listing appears in Figure 3.10.



```
BREAK-EVEN POINT

 FIXED COSTS ($)        1500
 COST PER UNIT ($)      3
 SALES PRICE ($)        6

BREAK-EVEN POINT  >     500 UNITS









-----------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN Y OR N ?█
```

Figure 3.9: Screen Display: Break-Even Point

```
980 REM ------------------------------
990 N$="BREAK-EVEN POINT"
1000 REM ------------------------------
1010 GRAPHICS 0:PRINT N$:PRINT :K=1
1020 REM GET INPUT PARAMETERS
1030 Q1$="ENTER DATA":Q2$=""
1040 Q3$="FIXED COSTS ($)":GOSUB 6600
1050 Q3$="COST PER UNIT ($)":GOSUB 6600
1060 Q3$="SALES PRICE ($)":GOSUB 6600
1070 PRINT
1080 PRINT "BREAK-EVEN POINT ->   ";
1090 US=INT(P(1)/(P(3)-P(2))*100+0.5)/100
1100 US=INT(US*100+0.5)/100
1110 PRINT US;" UNITS"
1120 Q2$="COMPUTE":GOSUB 3400
1130 IF YN$="N" THEN RETURN
1140 GOTO 980
```

Figure 3.10: Program Listing: Break-Even Point

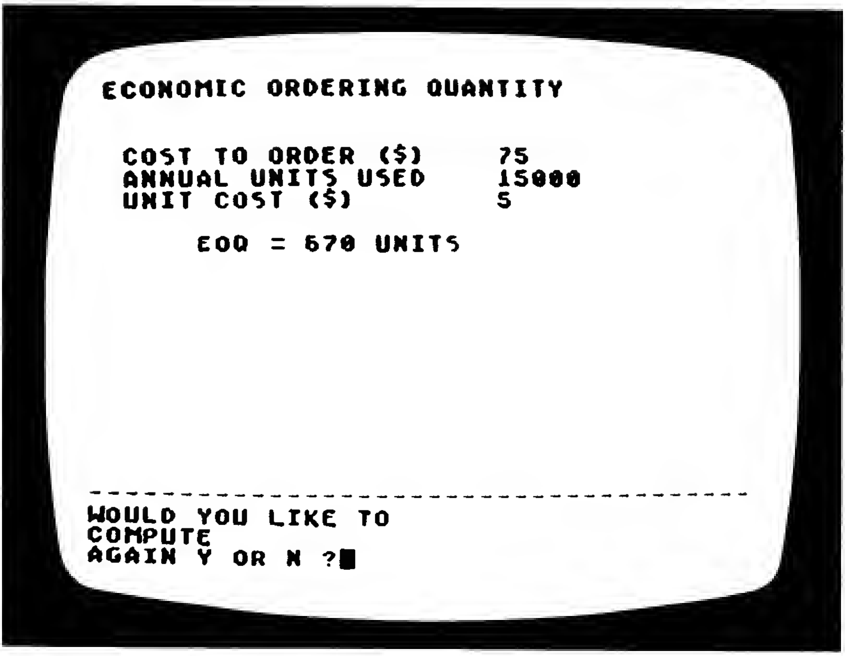
ECONOMIC ORDERING QUANTITY

Description

Operations researchers have developed a method to compute the most economic quantity of items for a manufacturer to order. This program does the computation for you. To use this program, specify the cost to place a purchase order, the number of units you use annually, and the annual carrying cost per unit. Carrying cost is defined as the interest rate your funds earn multiplied by the purchase price per item.

Example

Ray is the production manager for a small manufacturing company. He wants to calculate the economic ordering quantity for pumps that are used in the company's products. The company uses 15,000 pumps annually. The annual carrying cost of the pumps is $5 each, and a purchase order costs $75. What is the economic ordering quantity? Figure 3.11 shows the result of Ray's analysis. Figure 3.12 displays the program listing.



```
ECONOMIC ORDERING QUANTITY

  COST TO ORDER ($)     75
  ANNUAL UNITS USED     15000
  UNIT COST ($)         5

        EOQ = 670 UNITS





-------------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN Y OR N ?█
```

Figure 3.11: Screen Display: Economic Ordering Quantity

```
1150 REM ------------------------------
1160 N$="ECONOMIC ORDERING QUANTITY"
1170 REM ------------------------------
1180 GRAPHICS 0:PRINT N$:PRINT :K=1
1190 REM GET INPUT PARAMETERS
1200 Q1$="ENTER DATA":Q2$=""
1210 Q3$="COST TO ORDER ($)":GOSUB 6600
1220 Q3$="ANNUAL UNITS USED":GOSUB 6600
1230 Q3$="UNIT COST ($)":GOSUB 6600
1240 REM COMPUTE AND PRINT
1250 PRINT
1260 EOQ=INT(SQR(2*P(1)*P(2)/P(3)))
1270 PRINT "     EOQ = ";EOQ;" UNITS"
1280 Q2$="COMPUTE":GOSUB 3400
1290 IF YN$="N" THEN RETURN
1300 GOTO 1150
```

Figure 3.12: Program Listing: Economic Ordering Quantity

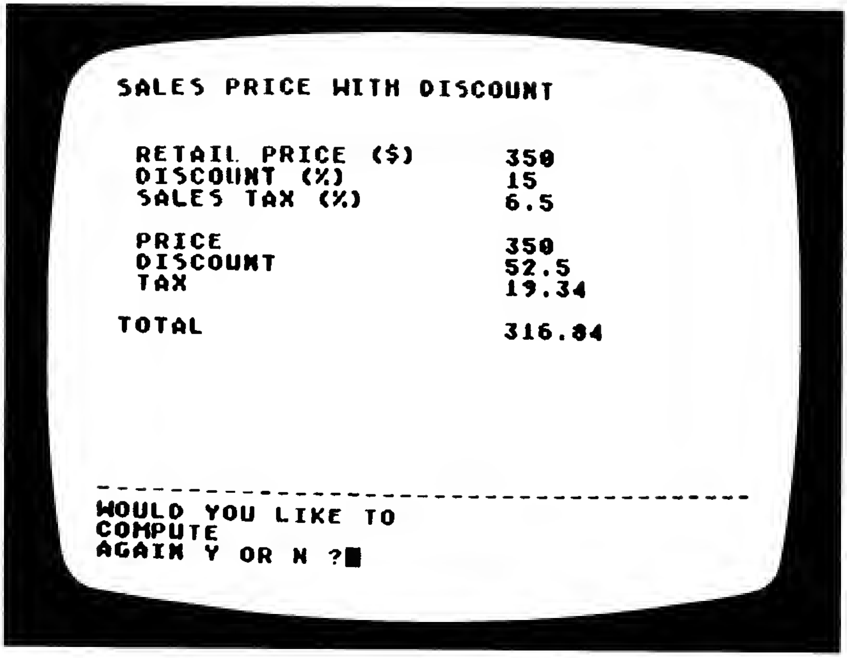
SALES PRICE WITH DISCOUNT

Description

Here is a program that enables you to compute the total sales price, with tax, of an item that is selling at a discount. To use this program you specify the retail price, discount percentage, and sales tax rate.

Example

Betty wants to buy a television set that is being sold at a 15% discount. The retail price of the television set is $350 and the sales tax is 6.5%. How much will Betty have to pay? Figure 3.13 shows the screen display for this example. Figure 3.14 shows the listing for the Sales Price with Discount program.



```
SALES PRICE WITH DISCOUNT

 RETAIL PRICE ($)          350
 DISCOUNT (%)              15
 SALES TAX (%)             6.5

 PRICE                     350
 DISCOUNT                  52.5
 TAX                       19.34

TOTAL                      316.84





----------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN Y OR N ?█
```

Figure 3.13: Screen Display: Sales Price with Discount

```
1310 REM -----------------------------
1320 N$="SALES PRICE WITH DISCOUNT"
1330 REM -----------------------------
1340 GRAPHICS 0:PRINT N$:PRINT :K=1
1350 REM GET INPUT PARAMETERS
1360 Q1$="ENTER DATA":Q2$=""
1370 Q3$="RETAIL PRICE ($)":GOSUB 6600
1380 Q3$="DISCOUNT (%)":GOSUB 6600
1390 Q3$="SALES TAX (%)":GOSUB 6600
1400 REM COMPUTE AND PRINT
1410 PRINT
1420 PRINT " PRICE":POSITION 24,7:PRINT P(1)
1430 D=INT(P(1)*P(2)+0.5)/100
1440 PRINT " DISCOUNT":POSITION 24,8:PRINT D
1450 T=INT((P(1)-D)*P(3)+0.5)/100
1460 PRINT " TAX":POSITION 24,9:PRINT T
1470 PRINT
1480 PRINT "TOTAL":POSITION 24,11:PRINT P(1)-D+T
1490 Q2$="COMPUTE":GOSUB 3400
1500 IF YN$="N" THEN RETURN
1510 GOTO 1310
```

Figure 3.14: Program Listing: Sales Price with Discount

WEIGHTED AVERAGE

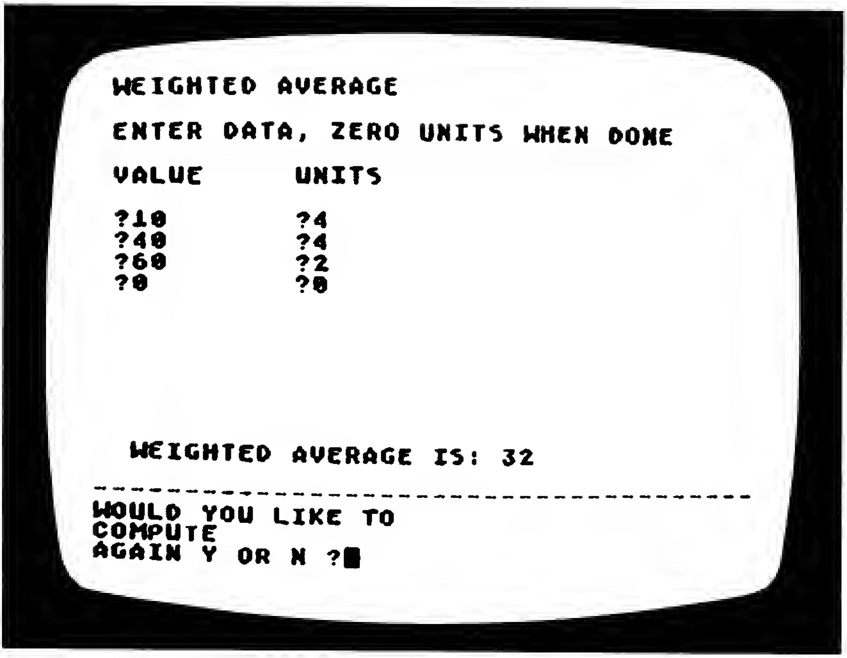
Description

There are many times when you will need to compute the weighted average of a number of items. Weighted averages can be used to find the average cost of stocks when you buy shares of stock at different times and different values. Weighted averages are also one way of finding the value of a business's inventory when the same item is bought at different prices.

This program rapidly computes the weighted average of a series of quantities. You use the program by entering the cost per unit followed by the number of units purchased for each item. You end the entry process by typing 0 for the number of units.

Example

Lloyd has been buying stock in a rapidly appreciating electronics company. He would like to calculate his average cost of shares; he has purchased four shares at $10 each, four shares at $40 each, and two shares at $60 each. Figures 3.15 and 3.16 show the result of the calculation and the program listing.



```
WEIGHTED AVERAGE

ENTER DATA, ZERO UNITS WHEN DONE

VALUE       UNITS

?10         ?4
?40         ?4
?60         ?2
?0          ?0



 WEIGHTED AVERAGE IS: 32
--------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN Y OR N ?
```

Figure 3.15: Screen Display: Weighted Average

```
1520 REM ------------------------------
1530 N$="WEIGHTED AVERAGE"
1540 REM ------------------------------
1550 GRAPHICS 0:PRINT N$:PRINT :K=1
1560 REM GET INPUT PARAMETERS
1570 PRINT "ENTER DATA, ZERO UNITS WHEN DONE"
1580 PRINT :PRINT "VALUE     UNITS"
1590 AVE=0:U=0:ND=0
1600 ND=ND+1:L=ND+5-INT((ND-1)/10)*10
1610 POSITION 1,L:PRINT S$:POSITION 2,L
1620 INPUT VA:POSITION 12,L:INPUT UN
1630 IF UN=0 THEN 1670
1640 AVE=AVE+VA*UN
1650 U=U+UN
1660 GOTO 1600
1670 AVE=AVE/U
1680 POSITION 3,17
1690 PRINT "WEIGHTED AVERAGE IS: ";AVE
1700 Q2$="COMPUTE":GOSUB 3400:REM ASK FOR ANOTHER
1710 IF YN$="N" THEN RETURN
1720 GOTO 1520
```

Figure 3.16: Program Listing: Weighted Average

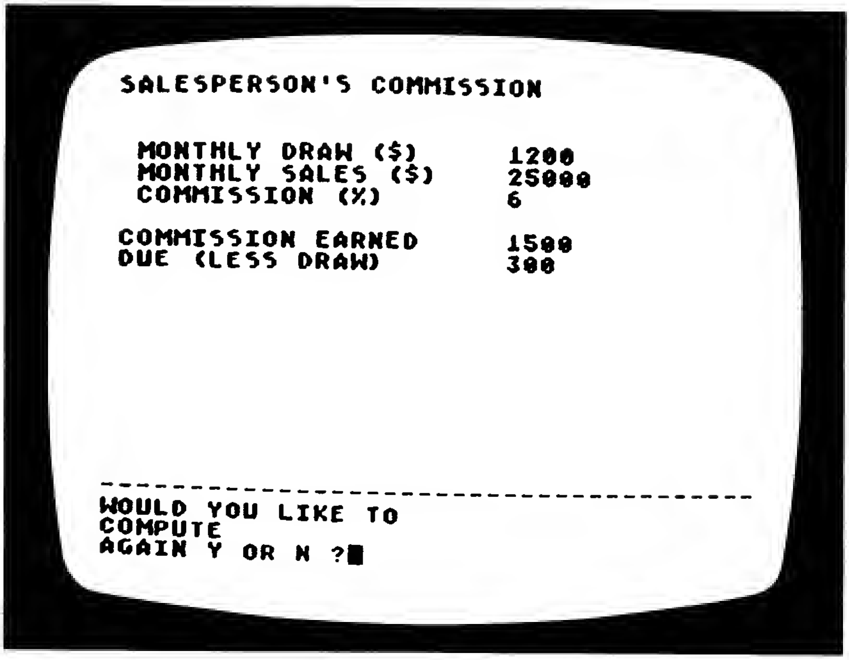
SALESPERSON'S COMMISSION

Description

We will now enter a program to calculate the commission due to a salesperson. You specify the salesperson's monthly draw, monthly sales volume, and commission percentage. A negative result indicates that the salesperson has not sold his quota.

Example

Sam Spade receives a monthly draw of $1,200. He earns 6% commission on the sales he makes as a furniture salesman. What commission is owed him for a month in which he sells $25,000 worth of furniture? Figures 3.17 and 3.18 are the screen display and the program listing for the Salesperson's Commission program.



```
SALESPERSON'S COMMISSION

 MONTHLY DRAW ($)        1200
 MONTHLY SALES ($)       25000
 COMMISSION (%)          6

COMMISSION EARNED        1500
DUE (LESS DRAW)          300






--------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN Y OR N ?█
```

Figure 3.17: Screen Display: Salesperson's Commission

```
1730 REM ------------------------------
1740 N$="SALESPERSON'S COMMISSION"
1750 REM ------------------------------
1760 GRAPHICS 0:PRINT N$:PRINT :K=1
1770 REM GET INPUT PARAMETERS
1780 Q1$="ENTER DATA":Q2$=""
1790 Q3$="MONTHLY DRAW ($)":GOSUB 6600
1800 Q3$="MONTHLY SALES ($)":GOSUB 6600
1810 Q3$="COMMISSION (%)":GOSUB 6600
1820 REM COMPUTE AND PRINT
1830 PRINT
1840 C=INT(P(2)*P(3)+0.5)/100
1850 PRINT "COMMISSION EARNED":POSITION 24,7:PRINT C
1860 D=INT((P(2)*P(3)/100-P(1))*100+0.5)/100
1870 PRINT "DUE (LESS DRAW)":POSITION 24,8:PRINT D
1880 Q2$="COMPUTE":GOSUB 3400
1890 IF YN$="N" THEN RETURN
1900 GOTO 1730
```

Figure 3.18: Program Listing: Salesperson's Commission

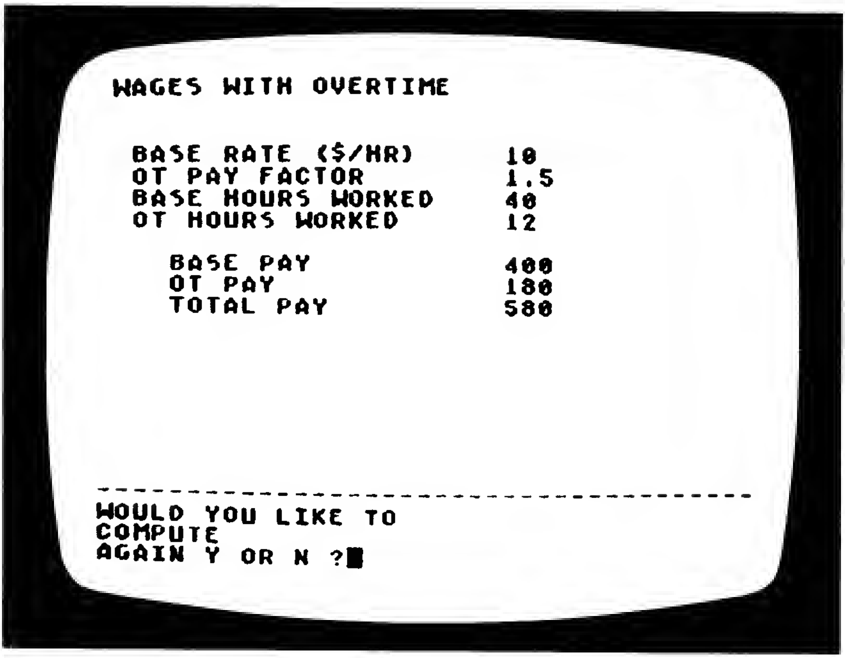
WAGES WITH OVERTIME

Description

This program makes it easy to calculate the total wages due an employee who earns overtime pay at a premium rate. You enter the employee's base hourly pay, the overtime pay factor, and the number of base and overtime hours that the employee worked.

Example

Toni earns $10.00 an hour as a machine operator. During a very busy week she works her regular 40 hours plus 12 hours at overtime pay. You wish to compute her total wages, with overtime paid at time and a half. Figures 3.19 and 3.20 show the results of the overtime computation and the program listing.



```
WAGES WITH OVERTIME

 BASE RATE ($/HR)       10
 OT PAY FACTOR          1.5
 BASE HOURS WORKED      40
 OT HOURS WORKED        12

   BASE PAY             400
   OT PAY               180
   TOTAL PAY            580




-------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN Y OR N ?
```

Figure 3.19: Screen Display: Wages with Overtime

```
1910 REM ------------------------------
1920 N$="WAGES WITH OVERTIME"
1930 REM ------------------------------
1940 GRAPHICS 0:PRINT N$:PRINT :K=1
1950 REM GET INPUT PARAMETERS
1960 Q1$="ENTER DATA":Q2$=""
1970 Q3$="BASE RATE ($/HR)":GOSUB 6600
1980 Q3$="OT PAY FACTOR":GOSUB 6600
1990 Q3$="BASE HOURS WORKED":GOSUB 6600
2000 Q3$="OT HOURS WORKED":GOSUB 6600
2010 REM COMPUTE AND PRINT
2020 PRINT
2030 BP=INT(P(1)*P(3)*100+0.5)/100
2040 PRINT "   BASE PAY":POSITION 24,8:PRINT BP
2050 OP=INT(P(1)*P(2)*P(4)*100+0.5)/100
2060 PRINT "   OT PAY":POSITION 24,9:PRINT OP
2070 PRINT "   TOTAL PAY":POSITION 24,10:PRINT OP+BP
2080 Q2$="COMPUTE":GOSUB 3400
2090 IF YN$="N" THEN RETURN
2100 GOTO 1910
```

Figure 3.20: Program Listing: Wages with Overtime

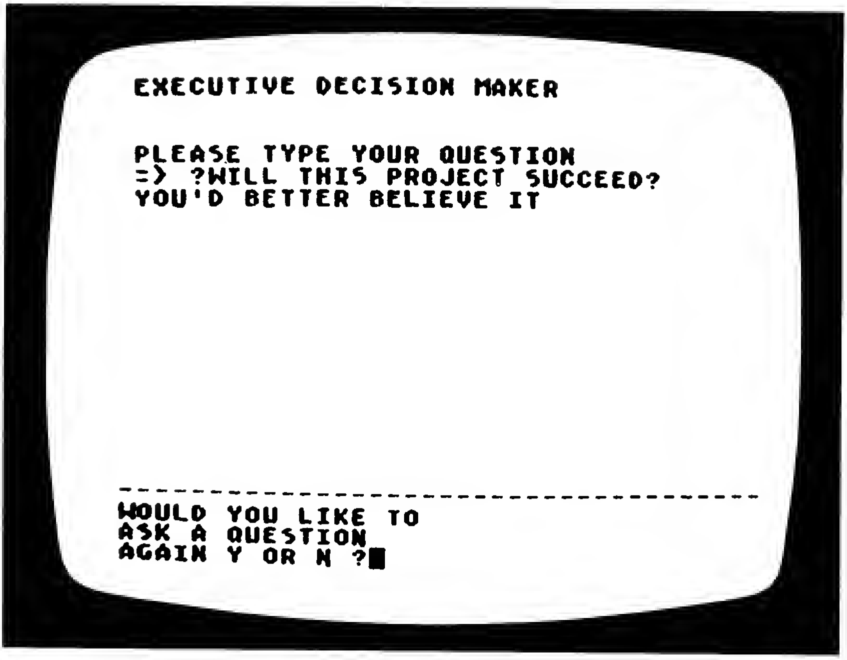
EXECUTIVE DECISION MAKER

Description

When all else fails, you can use this program to assist you in making key business decisions. Simply type your question into the computer. The program uses advanced decision theory to answer your question. (This program is fun to run at a cocktail party or other social gatherings.)

Example

Ed needs to know whether a high-risk project will succeed. Because his staff members cannot provide him with a definite recommendation, he asks his computer, as shown in Figure 3.21. The listing for this program is in Figure 3.22.



```
EXECUTIVE DECISION MAKER

PLEASE TYPE YOUR QUESTION
=> ?WILL THIS PROJECT SUCCEED?
YOU'D BETTER BELIEVE IT






-------------------------------------
WOULD YOU LIKE TO
ASK A QUESTION
AGAIN Y OR N ?█
```

Figure 3.21: Screen Display: Executive Decision Maker

```
2110 REM -----------------------------
2120 N$="EXECUTIVE DECISION MAKER"
2130 REM -----------------------------
2140 GRAPHICS 0:PRINT N$:PRINT :K=1
2150 PRINT
2160 PRINT "PLEASE TYPE YOUR QUESTION"
2170 PRINT "=> ";
2180 INPUT Q1$: Y=LEN(Q1$)
2190 IF Y<6 THEN 2210
2200 Y=INT(Y/2):GOTO 2190
2210 ON Y GOTO 2220,2230,2240,2250,2260
2220 PRINT "NOT ON YOUR LIFE":GOTO 2270
2230 PRINT "GO FOR IT!!!":GOTO 2270
2240 PRINT "YOU'D BETTER BELIEVE IT":GOTO 2270
2250 PRINT "FLIP A COIN!":GOTO 2270
2260 PRINT "MAKE YOUR BREAK- QUICKLY"
2270 Q2$="ASK A QUESTION":GOSUB 3400
2280 IF YN$="N" THEN RETURN
2290 GOTO 2110
```

Figure 3.22: Program Listing: Executive Decision Maker

4
REAL ESTATE PROGRAMS

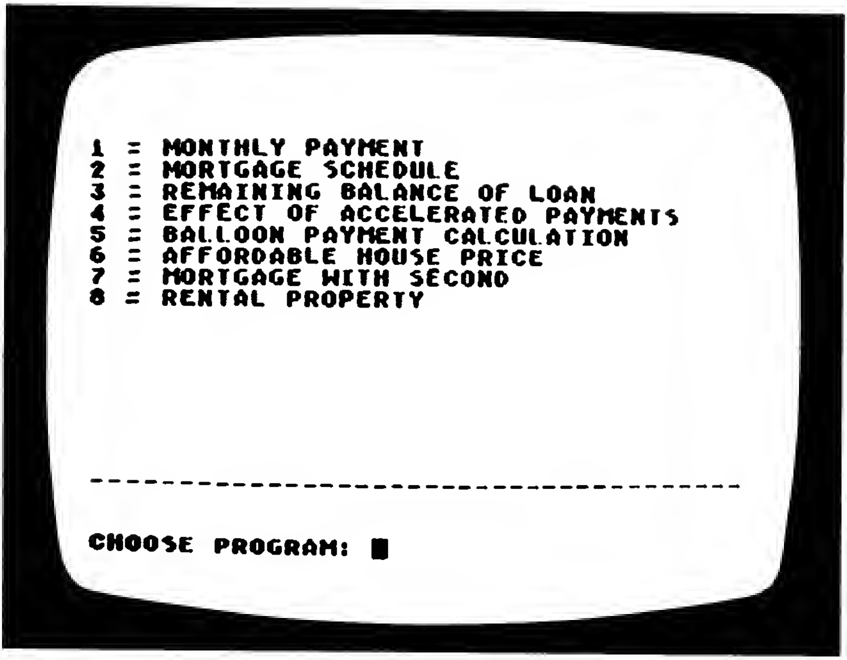
In this chapter we present eight useful real estate programs. These programs allow you to set up mortgage payment schedules and perform many other practical computations.

Two programs in this chapter are especially interesting. You can use the Affordable House Price program to find the sales price of a home that a bank will finance. This program is helpful both in finding a price range you can afford, and, should you decide to purchase, in negotiating a price. Another program you will find particularly enlightening is the Accelerated Payments program. A small increase in your monthly payment can drastically reduce the total interest you pay on a home. Take a look at Figure 4.10, and you'll see what we mean!

As in the other chapters, we provide a menu selection program that you can use to create a set of real estate programs.

REAL ESTATE PROGRAMS MENU

We can again combine the programs in this chapter to provide a menu-driven real estate analysis package. Instructions for saving the menu program and combining it with the other programs in this chapter are in Chapter 1. The menu display and its program listing are shown in Figures 4.1 and 4.2.



```
1 = MONTHLY PAYMENT
2 = MORTGAGE SCHEDULE
3 = REMAINING BALANCE OF LOAN
4 = EFFECT OF ACCELERATED PAYMENTS
5 = BALLOON PAYMENT CALCULATION
6 = AFFORDABLE HOUSE PRICE
7 = MORTGAGE WITH SECOND
8 = RENTAL PROPERTY




--------------------------------------

CHOOSE PROGRAM: █
```

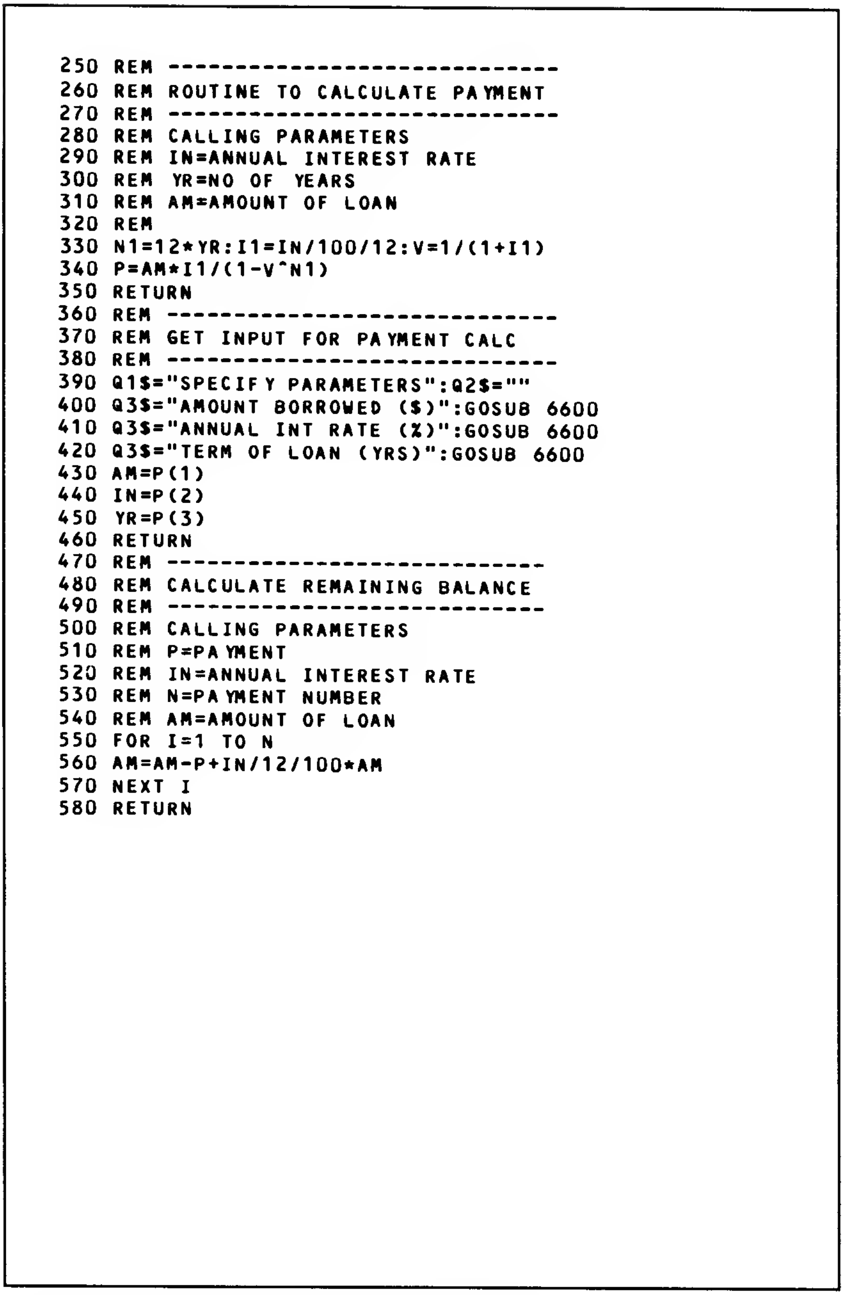
Figure 4.1: Screen Display: Real Estate Programs Menu

```
100 REM --------------------------
110 REM REAL ESTATE PROGRAMS
120 REM --------------------------
130 GOSUB 6200:REM INITIALIZE
140 X1$="MONTHLY PAYMENT"
150 X2$="MORTGAGE SCHEDULE"
160 X3$="REMAINING BALANCE OF LOAN"
170 X4$="EFFECT OF ACCELERATED PAYMENTS"
180 X5$="BALLOON PAYMENT CALCULATION"
190 X6$="AFFORDABLE HOUSE PRICE"
200 X7$="MORTGAGE WITH SECOND"
210 X8$="RENTAL PROPERTY"
220 N=8:GOSUB 7000:REM DISPLAY MENU
230 ON X GOSUB 590,710,910,1070,,1470,1650,1890,2170
240 GOTO 140
```

Figure 4.2: Program Listing: Real Estate Programs Menu

REAL ESTATE SUBROUTINES

The programs in this chapter make use of three common subroutines. You should first enter these subroutines and save them on disk. When you are ready to enter a program, load the common subroutines. Then type in the program and save the subroutines and program together as a new disk file. (Merge SUBLIB with this file prior to running your program.) You should enter all the programs in the chapter on the same disk file with these subroutines. The common subroutines are shown in Figure 4.3. These routines are used to get input for a payment calculation, calculate a payment, and compute the remaining balance of a loan.



```
250 REM ----------------------------
260 REM ROUTINE TO CALCULATE PAYMENT
270 REM ----------------------------
280 REM CALLING PARAMETERS
290 REM IN=ANNUAL INTEREST RATE
300 REM YR=NO OF YEARS
310 REM AM=AMOUNT OF LOAN
320 REM
330 N1=12*YR:I1=IN/100/12:V=1/(1+I1)
340 P=AM*I1/(1-V^N1)
350 RETURN
360 REM ----------------------------
370 REM GET INPUT FOR PAYMENT CALC
380 REM ----------------------------
390 Q1$="SPECIFY PARAMETERS":Q2$=""
400 Q3$="AMOUNT BORROWED ($)":GOSUB 6600
410 Q3$="ANNUAL INT RATE (%)":GOSUB 6600
420 Q3$="TERM OF LOAN (YRS)":GOSUB 6600
430 AM=P(1)
440 IN=P(2)
450 YR=P(3)
460 RETURN
470 REM ---------------------------
480 REM CALCULATE REMAINING BALANCE
490 REM ---------------------------
500 REM CALLING PARAMETERS
510 REM P=PAYMENT
520 REM IN=ANNUAL INTEREST RATE
530 REM N=PAYMENT NUMBER
540 REM AM=AMOUNT OF LOAN
550 FOR I=1 TO N
560 AM=AM-P+IN/12/100*AM
570 NEXT I
580 RETURN
```

Figure 4.3: Program Listing: Real Estate Subroutines

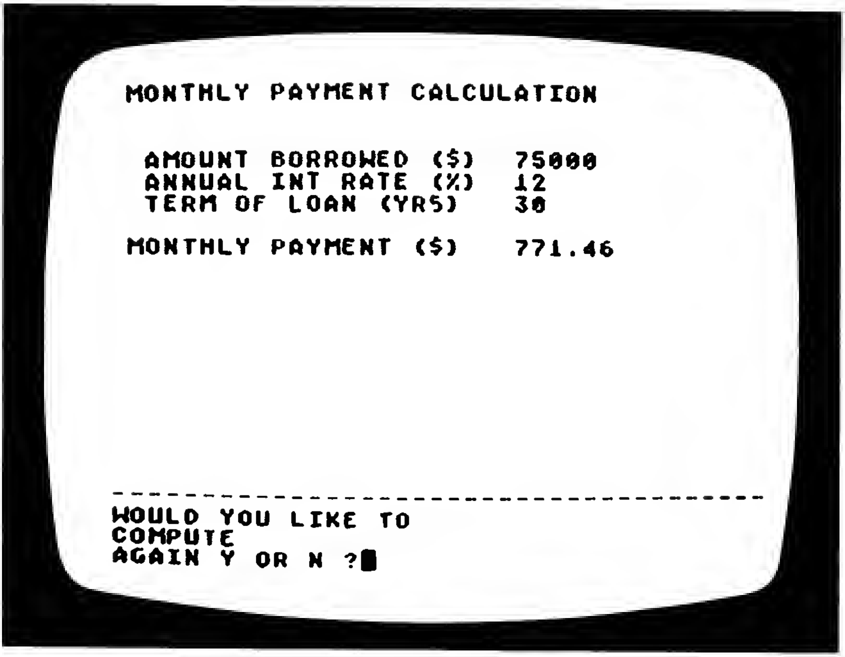
MONTHLY PAYMENT CALCULATION

Description

This program calculates the monthly payment necessary to fully amortize a loan. You must provide the amount of the loan, the annual interest rate, and the term of the loan in years.

Example

Don and Donna can purchase a house for $100,000. They plan to make a $25,000 down payment and finance the balance at 12% for 30 years. Find their monthly payment. Figures 4.4 and 4.5 show the screen display and the program listing for the Monthly Payment program.



```
MONTHLY PAYMENT CALCULATION

 AMOUNT BORROWED ($)   75000
 ANNUAL INT RATE (%)   12
 TERM OF LOAN (YRS)    30

MONTHLY PAYMENT ($)    771.46





-----------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN Y OR N ?
```

Figure 4.4: Screen Display: Monthly Payment Calculation

```
590 REM -----------------------------
600 N$="MONTHLY PAYMENT CALCULATION"
610 REM -----------------------------
620 GRAPHICS 0:PRINT N$:PRINT :K=1
630 GOSUB 360:REM GET PARAMETERS
640 GOSUB 250:REM CALCULATE PMT
650 PRINT
660 P=INT(P*100+0.5)/100
670 PRINT "MONTHLY PAYMENT ($)":POSITION 24,7:PRINT P
680 Q2$="COMPUTE":GOSUB 3400:REM ASK FOR ANOTHER
690 IF YN$="N" THEN RETURN
700 GOTO 590
```

Figure 4.5: Program Listing: Monthly Payment Calculation

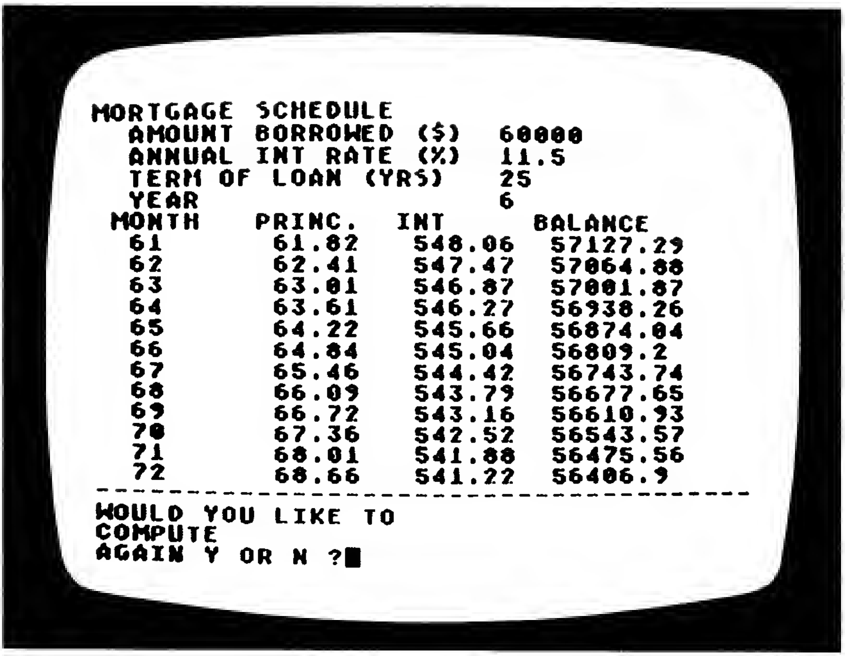
MORTGAGE SCHEDULE

Description

This program displays a mortgage schedule on the screen for any year you choose. You enter the amount of the loan, the annual interest rate, the term of the loan, and the year for which you wish the mortgage schedule to be displayed. Thus, you can rapidly review the principal and interest you pay in any year. The interest you pay is deductible from your income tax.

Example

The mortgage on Sam and Samantha's home is $60,000. The annual interest rate is 11.5%, and the term of the loan is 25 years. Sam and Samantha would like to review the principal and interest they will be paying during the sixth year of the loan. Figure 4.6 shows the screen display for this example. The program listing is displayed in Figure 4.7.



```
MORTGAGE SCHEDULE
  AMOUNT BORROWED ($)  60000
  ANNUAL INT RATE (%)  11.5
  TERM OF LOAN (YRS)   25
  YEAR                 6
 MONTH   PRINC.  INT     BALANCE
  61      61.82   548.06  57127.29
  62      62.41   547.47  57064.88
  63      63.01   546.87  57001.87
  64      63.61   546.27  56938.26
  65      64.22   545.66  56874.04
  66      64.84   545.04  56809.2
  67      65.46   544.42  56743.74
  68      66.09   543.79  56677.65
  69      66.72   543.16  56610.93
  70      67.36   542.52  56543.57
  71      68.01   541.88  56475.56
  72      68.66   541.22  56406.9
---------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN Y OR N ?█
```

Figure 4.6: Screen Display: Mortgage Schedule

```
710 REM ----------------------------
720 N$="MORTGAGE SCHEDULE"
730 REM ----------------------------
740 GRAPHICS 0:PRINT N$:PRINT :K=1
750 GOSUB 360:REM GET LOAN PARAMETERS
760 Q3$="YEAR"
770 GOSUB 6600:N=P(4):REM GET YEAR
780 GOSUB 250:REM CALCULATE PAYMENT
790 N=12*N-12:REM COMPUTE FIRST MONTH
800 GOSUB 470:REM GET BEGINNING BALANCE
810 PRINT "MONTH   PRINC.  INT     BALANCE"
820 FOR J=1 TO 12
830 I1=IN/12/100*AM:P1=P-I1:AM=AM-P1
840 N=N+1:P1=INT(P1*100+0.5)/100
845 POSITION 1,J+7:PRINT S$;"  "
850 I1=INT(I1*100+0.5)/100:AM=INT(AM*100+0.5)/100
855 POSITION 3,J+7:PRINT N;:POSITION 11,J+7:PRINT P1
860 POSITION 19,J+7:PRINT I1:POSITION 27,J+7:PRINT AM
870 NEXT J
875 PRINT :PRINT :PRINT :PRINT :POSITION 1,1:PRINT N$
880 Q2$="COMPUTE":GOSUB 3400
890 IF YN$="N" THEN RETURN
900 GOTO 710
```

Figure 4.7: Program Listing: Mortgage Schedule

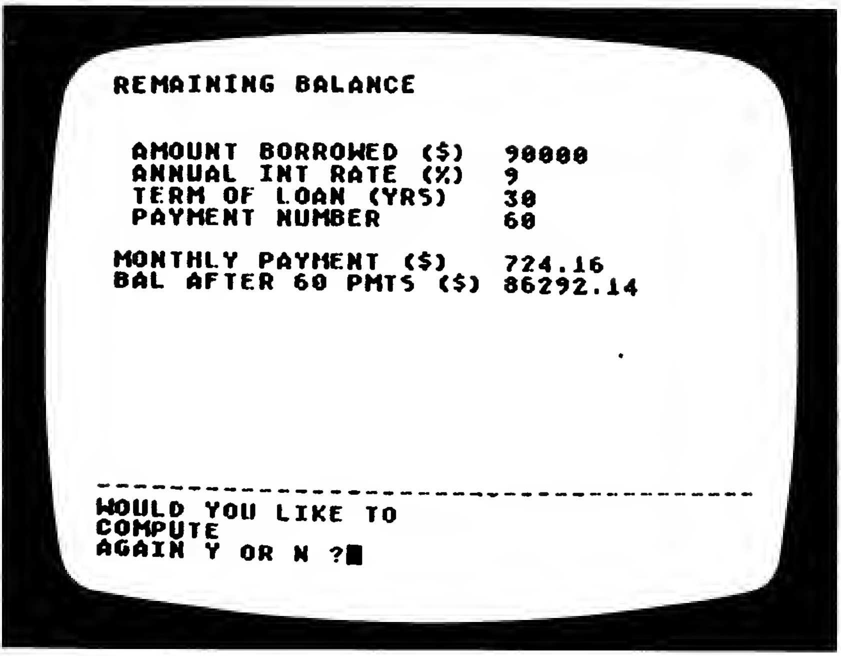
REMAINING BALANCE OF A LOAN

Description

This program finds the remaining balance on a loan. You can run this program to calculate the amount due should you choose to pay off the loan. The program assumes that all prior payments have been the exact monthly payment. To use the program you specify the amount of the mortgage, the interest rate and length of the loan, and the payment number for which you want to determine the remaining balance. Don't be concerned if this program takes a while to display its answer.

Example

John's mortgage is $90,000. His loan is for 30 years at an annual interest rate of 9%. He would like to compute the loan balance at the end of five years (the sixtieth payment). Figures 4.8 and 4.9 show the screen display and the program listing for the Remaining Balance program.



```
REMAINING BALANCE

 AMOUNT BORROWED ($)   90000
 ANNUAL INT RATE (%)   9
 TERM OF LOAN (YRS)    30
 PAYMENT NUMBER        60

MONTHLY PAYMENT ($)    724.16
BAL AFTER 60 PMTS ($)  86292.14



-------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN Y OR N ?█
```

Figure 4.8: Screen Display: Remaining Balance of a Loan

```
910 REM -------------------------
920 N$="REMAINING BALANCE"
930 REM -------------------------
940 GRAPHICS 0:PRINT N$:PRINT :K=1
950 GOSUB 360:REM GET LOAN PARAMETERS
960 Q3$="PAYMENT NUMBER"
970 GOSUB 6600:N=P(4):REM GET PMT NO.
980 GOSUB 250:REM CALCULATE PAYMENT
990 GOSUB 470:REM COMPUTE BALANCE
1000 PRINT
1010 P=INT(P*100+0.5)/100:AM=INT(AM*100+0.5)/100
1020 PRINT "MONTHLY PAYMENT ($)":POSITION 24,8:PRINT P
1030 PRINT "BAL AFTER ";N;" PMTS ($)"
1035 POSITION 24,9:PRINT AM
1040 Q2$="COMPUTE":GOSUB 3400
1050 IF YN$="N" THEN RETURN
1060 GOTO 910
```

Figure 4.9: Program Listing: Remaining Balance of a Loan

EFFECT OF ACCELERATED PAYMENTS

Description

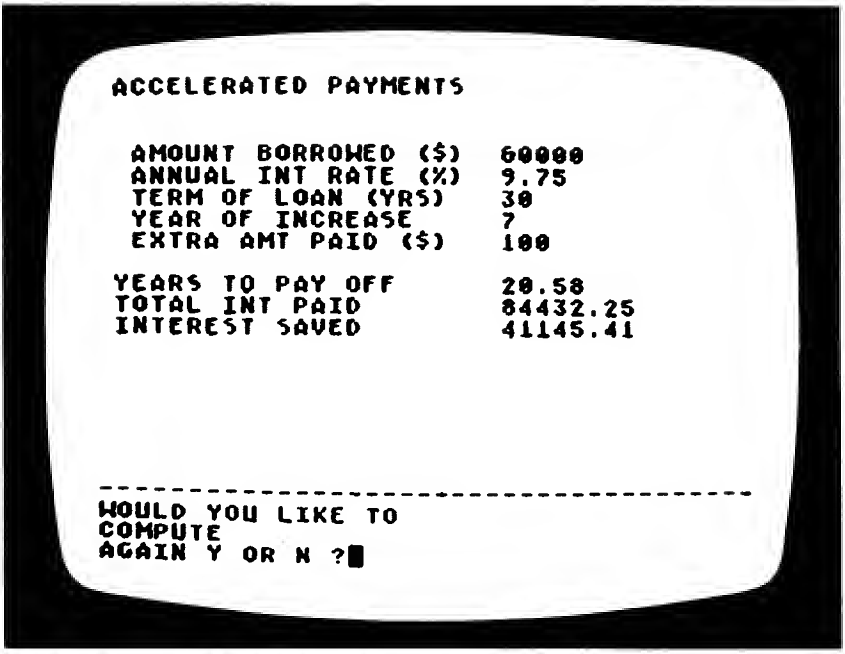
You will find this program extremely useful—it allows you to find the effect of increasing the monthly payments on your mortgage. An increase in the monthly payments will reduce both the number of years of payments and the total interest you pay. To use this program, enter the amount of the loan, the annual interest rate, the length of the loan, the year in which you increase the payments, and the extra amount you will pay each month. This program computes as many times as necessary to reduce the loan balance to zero, so you may find that it takes a while for the answer to appear.

Example

Cindy has decided to increase the monthly payments on her home loan by $100 each month. She begins the increased payments in the seventh year of the loan and would like to know how much she will save on the interest. Her loan is for $60,000 at 9.75%; the term of her loan is 30 years. You can see in Figure 4.10 that Cindy will save $41,000 with this increase in her monthly payment, and she will reduce the term of the loan by about 9½ years. Figure 4.11 shows the program listing for the Accelerated Payments program.

```
1070 REM -----------------------
1080 N$="ACCELERATED PAYMENTS"
1090 REM -----------------------
1100 GRAPHICS 0:PRINT N$:PRINT :K=1
1110 GOSUB 360:REM GET LOAN PARAMETERS
1120 Q3$="YEAR OF INCREASE"
1130 GOSUB 6600:N=P(4):REM GET YEAR
1140 GOSUB 250:REM CALCULATE PAYMENT
1150 N=12*N-12:REM COMPUTE FIRST MONTH
1160 Q3$="EXTRA AMT PAID ($)"
1170 GOSUB 6600:EX=P(5):REM GET EXTRA
1180 REM COMPUTE PRIN INT AND BAL TO DATE
1190 TI=0:BAL=0:TP=0
1200 FOR J=1 TO N
1210 I1=IN/12/100*AM:P1=P-I1:AM=AM-P1
1220 TI=TI+I1:TP=TP+P1
1230 NEXT J
```

Figure 4.11: Program Listing: Effect of Accelerated Payments (continues)



```
ACCELERATED PAYMENTS

 AMOUNT BORROWED ($)  60000
 ANNUAL INT RATE (%)  9.75
 TERM OF LOAN (YRS)   30
 YEAR OF INCREASE     7
 EXTRA AMT PAID ($)   100

YEARS TO PAY OFF      20.58
TOTAL INT PAID        84432.25
INTEREST SAVED        41145.41




--------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN Y OR N ?
```

Figure 4.10: Screen Display: Effect of Accelerated Payments

```
1240 REM COMPUTE INTEREST WITH NO INCREASE
1250 TI2=0:AM2=P(1)
1260 FOR J=1 TO 12*YR
1270 I2=IN/12/100*AM2:P2=P-I2:AM2=AM2-P2
1280 TI2=TI2+I2
1290 NEXT J
1300 REM COMPUTE WITH EXTRA PAYMENT
1310 J=0
1320 I1=IN/12/100*AM:TI=TI+I1
1330 P1=P+EX-I1:TP=TP+P1
1340 AM=AM-P1
1350 J=J+1
1360 IF AM>0 THEN 1320
1370 PRINT
1380 YR=INT((P(4)+J/12-1)*100+0.5)/100
1390 PRINT "YEARS TO PAY OFF":POSITION 24,9:PRINT YR
1400 TI=INT(TI*100+0.5)/100
1410 PRINT "TOTAL INT PAID":POSITION 24,10:PRINT TI
1420 TI2=INT((TI2-TI)*100+0.5)/100
1430 PRINT "INTEREST SAVED":POSITION 24,11:PRINT TI2
1440 Q2$="COMPUTE":GOSUB 3400
1450 IF YN$="N" THEN RETURN
1460 GOTO 1070
```

Figure 4.11: Program Listing: Effect of Accelerated Payments

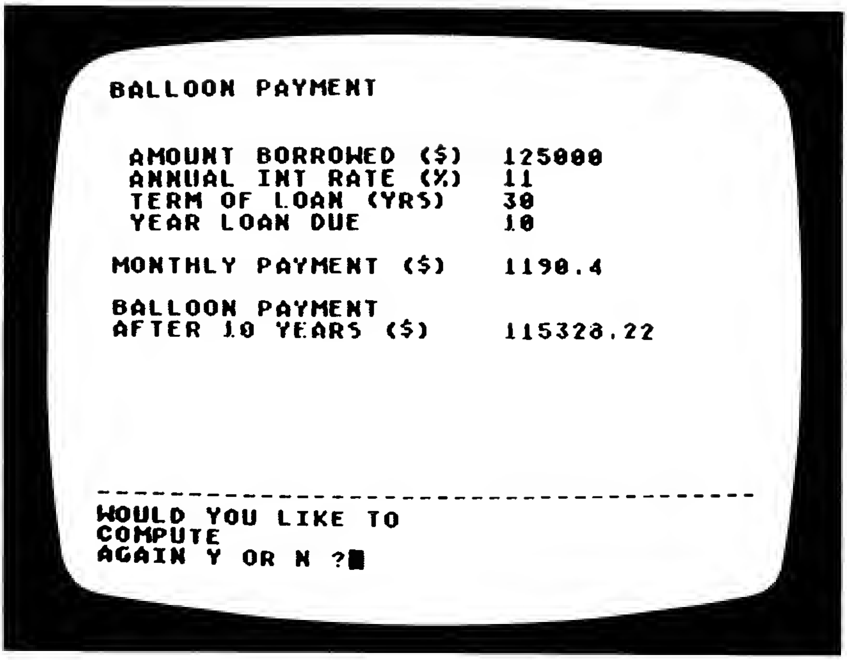
BALLOON PAYMENT CALCULATION

Description

Many recent mortgages have been set up with a balloon payment due at the end of a specified period. This program allows you to calculate the required balloon payment. You must enter the amount of the loan, the annual interest rate, the term of the loan, and the year in which the balloon payment is due.

Example

Max has a mortgage for $125,000. The annual interest rate is 11%. The payments are amortized over 30 years; Max must pay off the entire loan at the end of 10 years. What will be the amount of Max's balloon payment? Figures 4.12 and 4.13 show the screen display and the program listing for the Balloon Payment program.



```
BALLOON PAYMENT


 AMOUNT BORROWED ($)   125000
 ANNUAL INT RATE (%)   11
 TERM OF LOAN (YRS)    30
 YEAR LOAN DUE         10

MONTHLY PAYMENT ($)    1190.4

BALLOON PAYMENT
AFTER 10 YEARS ($)     115328.22







---------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN Y OR N ?
```

Figure 4.12: Screen Display: Balloon Payment Calculation

```
1470 REM -------------------------
1480 N$="BALLOON PAYMENT"
1490 REM -------------------------
1500 GRAPHICS 0:PRINT N$:PRINT :K=1
1510 GOSUB 360:REM GET LOAN PARAMETERS
1520 Q3$="YEAR LOAN DUE"
1530 GOSUB 6600:N=12*P(4):REM GET YEAR DUE
1540 GOSUB 250:REM CALCULATE PAYMENT
1550 GOSUB 470:REM COMPUTE BALANCE
1560 PRINT
1570 P=INT(P*100+0.5)/100
1580 PRINT "MONTHLY PAYMENT ($)":POSITION 24,8:PRINT P
1590 AM=INT(AM*100+0.5)/100
1600 PRINT :PRINT "BALLOON PAYMENT"
1610 PRINT "AFTER ";N/12;" YEARS ($)"
1615 POSITION 24,11:PRINT AM
1620 Q2$="COMPUTE":GOSUB 3400:REM ASK FOR ANOTHER
1630 IF YN$="N" THEN RETURN
1640 GOTO 1470
```

Figure 4.13: Program Listing: Balloon Payment Calculation

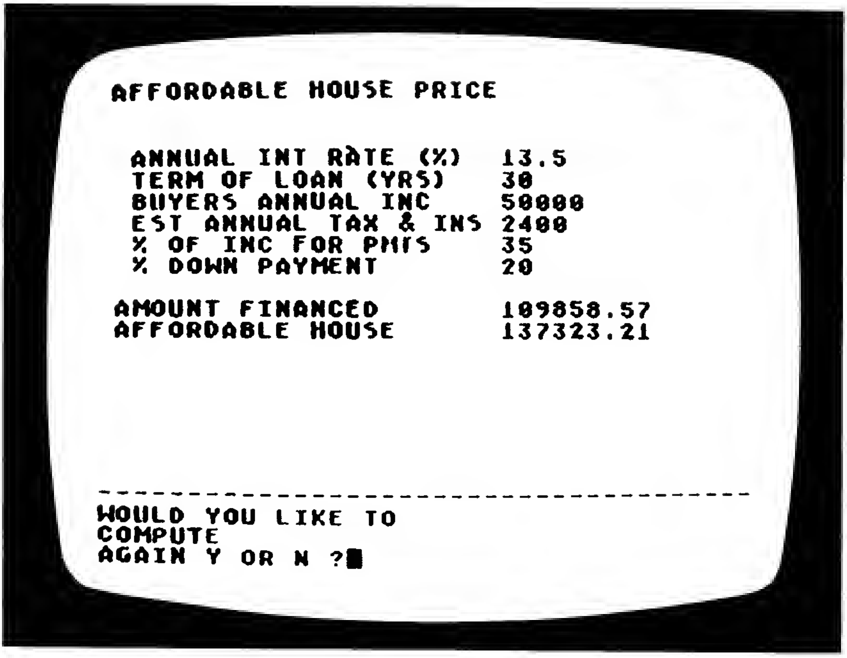
AFFORDABLE HOUSE PRICE

Description

You can use this program to find the price of a home for which a bank will give you a loan. This program can be invaluable when you are negotiating to buy a new home. To use the program, enter the annual interest rate and term of available loans, your annual income, the estimated annual taxes and insurance, the percent of your gross income that the bank will allow for your total monthly payments, and the percent down payment you will make.

Example

Bill and Betty would like to purchase a new home. The available interest rate for home loans is 13.5% for 30-year loans. The couple's annual income is $50,000 per year, and they plan to put 20% down on their new house. Bill and Betty estimate that their yearly taxes and insurance will be $2,400. Their banker has told them that the monthly payment on their house must not exceed 35% of their monthly income. Figures 4.14 and 4.15 are the screen display and the program listing for the Affordable House Price program.



```
AFFORDABLE HOUSE PRICE

 ANNUAL INT RATE (%)  13.5
 TERM OF LOAN (YRS)   30
 BUYERS ANNUAL INC    50000
 EST ANNUAL TAX & INS 2400
 % OF INC FOR PMTS    35
 % DOWN PAYMENT       20

AMOUNT FINANCED       109858.57
AFFORDABLE HOUSE      137323.21




---------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN Y OR N ?█
```

Figure 4.14: Screen Display: Affordable House Price

```
1650 REM -----------------------
1660 N$="AFFORDABLE HOUSE PRICE"
1670 REM -----------------------
1680 GRAPHICS 0:PRINT N$:PRINT :K=1
1690 Q1$="SPECIFY PARAMETERS":Q2$=""
1700 Q3$="ANNUAL INT RATE (%)":GOSUB 6600
1710 Q3$="TERM OF LOAN (YRS)":GOSUB 6600
1720 Q3$="BUYERS ANNUAL INC":GOSUB 6600
1730 Q3$="EST ANNUAL TAX & INS":GOSUB 6600
1740 Q3$="% OF INC FOR PMTS":GOSUB 6600
1750 Q3$="% DOWN PAYMENT":GOSUB 6600
1760 PRINT
1770 REM COMPUTE ALLOWABLE PAYMENT
1780 P=P(5)/100*P(3)/12-P(4)/12
1790 REM COMPUTE AMOUNT FINANCED
1800 IN=P(1)/100/12:N1=12*P(2):V=1/(1+IN)
1810 AM=P*(1-V^N1)/IN
1820 AM=INT(AM*100+0.5)/100
1830 AH=INT(AM/(1-P(6)/100)*100+0.5)/100
1840 PRINT "AMOUNT FINANCED":POSITION 24,10:PRINT AM
1850 PRINT "AFFORDABLE HOUSE":POSITION 24,11:PRINT AH
1860 Q2$="COMPUTE":GOSUB 3400
1870 IF YN$="N" THEN RETURN
1880 GOTO 1650
```

Figure 4.15: Program Listing: Affordable House Price

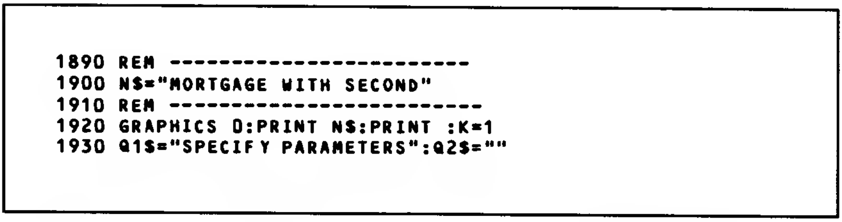
MORTGAGE WITH SECOND MORTGAGE

Description

We shall now examine a program you can use to find the total payments on a house that will have both a first and second mortgage. To use this program you enter the purchase price of the home, the cash available for a down payment, and the amount of the first mortgage you will obtain. You also specify the interest rate of the first and second mortgages and the length of the first. We assume that the second mortgage is paid monthly on an interest-only basis.

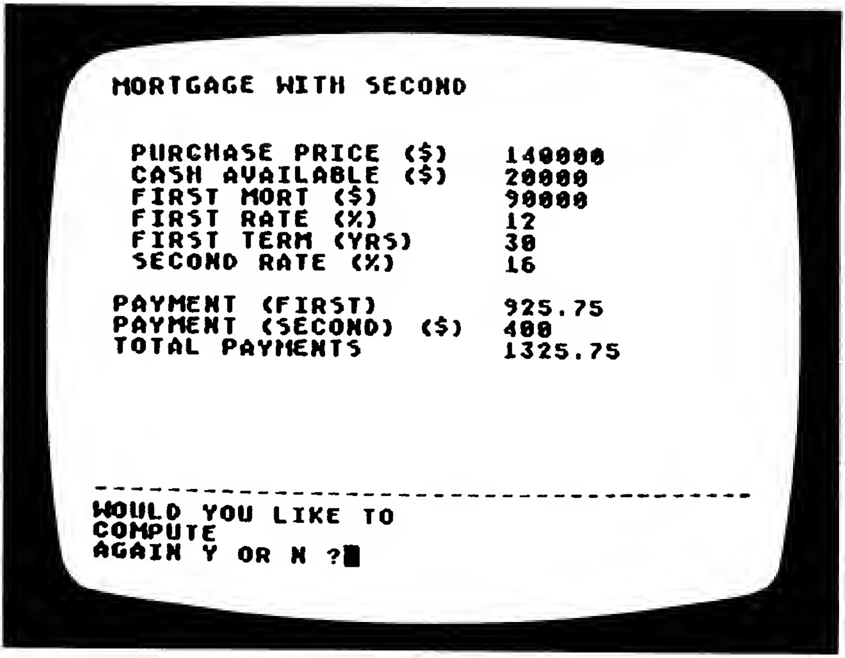
Example

Joanne has saved $20,000 for a down payment on a new house. The house will cost $140,000 and she plans to take out a first mortgage for $90,000. The first mortgage will be for 30 years at 12% annual interest. She will also have a 16% interest-only second mortgage. What will be her total payments? The resulting screen display for this example is shown in Figure 4.16; the program listing appears in Figure 4.17.



```
1890 REM ------------------------
1900 N$="MORTGAGE WITH SECOND"
1910 REM -------------------------
1920 GRAPHICS 0:PRINT N$:PRINT :K=1
1930 Q1$="SPECIF Y PARAMETERS":Q2$=""
```

Figure 4.17: Program Listing: Mortgage with Second Mortgage (continues)



```
MORTGAGE WITH SECOND

 PURCHASE PRICE ($)    140000
 CASH AVAILABLE ($)    20000
 FIRST MORT ($)        90000
 FIRST RATE (%)        12
 FIRST TERM (YRS)      30
 SECOND RATE (%)       16

PAYMENT (FIRST)        925.75
PAYMENT (SECOND) ($)   400
TOTAL PAYMENTS         1325.75



---------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN Y OR N ?
```

Figure 4.16: Screen Display: Mortgage with Second Mortgage

```
1940 Q3$="PURCHASE PRICE ($)":GOSUB 6600
1950 Q3$="CASH AVAILABLE ($)":GOSUB 6600
1960 Q3$="FIRST MORT ($)":GOSUB 6600
1970 Q3$="FIRST RATE (%)":GOSUB 6600
1980 Q3$="FIRST TERM (YRS)":GOSUB 6600
1990 Q3$="SECOND RATE (%)":GOSUB 6600
2000 PRINT
2010 AM=P(3):IN=P(4):YR=P(5)
2020 REM COMPUTE FIRST PAYMENT
2030 GOSUB 250
2040 P=INT(P*100+0.5)/100
2050 PRINT "PAYMENT (FIRST)":POSITION 24,10:PRINT P
2060 REM COMPUTE SECOND PAYMENT
2070 IF P(2)<P(1)-P(3) THEN 2100
2080 PRINT "SECOND MORTGAGE NOT REQUIRED"
2090 GOTO 2140
2100 P2=(P(1)-P(2)-P(3))*P(6)/100/12
2110 P2=INT(P2*100+0.5)/100
2120 PRINT "PAYMENT (SECOND) ($)":POSITION 24,11:PRINT P2
2130 PRINT "TOTAL PAYMENTS":POSITION 24,12:PRINT P+P2
2140 Q2$="COMPUTE":GOSUB 3400
2150 IF YN$="N" THEN RETURN
2160 GOTO 1890
```

Figure 4.17: Program Listing: Mortgage with Second Mortgage

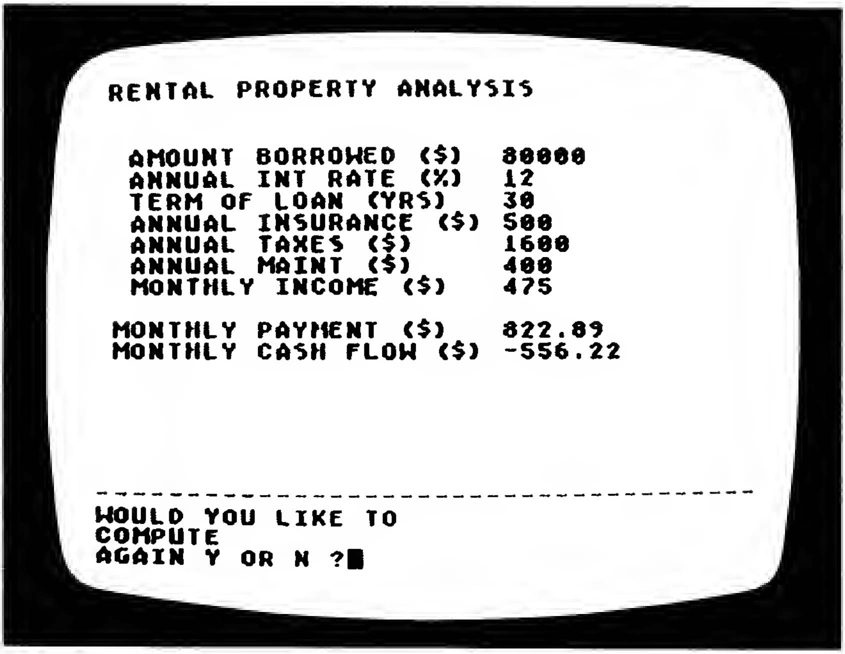
RENTAL PROPERTY ANALYSIS

Description

Let's now consider a program that is very useful for analyzing real estate investments. You can use the Rental Property Analysis program to find the monthly payment and cash flow on a rental property. (We don't consider the after-tax effects of depreciation.) You must enter the amount of the loan on your rental property, the annual interest rate and length of the loan, and the annual insurance, taxes, and maintenance. You must also enter the monthly rental income.

Example

Herb is planning to purchase a house, which he will rent for $475 per month. The first mortgage will be $80,000, amortized over 30 years at 12% interest. Herb estimates that the annual insurance will be $500, the property taxes $1,600, and the maintenance $400. What will be his monthly payment and cash flow? Figures 4.18 and 4.19 are the screen display and program listing for this example.



```
RENTAL PROPERTY ANALYSIS

 AMOUNT BORROWED ($)   80000
 ANNUAL INT RATE (%)   12
 TERM OF LOAN (YRS)    30
 ANNUAL INSURANCE ($)  500
 ANNUAL TAXES ($)      1600
 ANNUAL MAINT ($)      400
 MONTHLY INCOME ($)    475

MONTHLY PAYMENT ($)    822.89
MONTHLY CASH FLOW ($)  -556.22




---------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN Y OR N ?█
```

Figure 4.18: Screen Display: Rental Property Analysis

```
2170 REM -------------------------
2180 N$="RENTAL PROPERTY ANALYSIS"
2190 REM -------------------------
2200 GRAPHICS 0:PRINT N$:PRINT :K=1
2210 REM GET PARAMETERS
2220 GOSUB 360
2230 Q3$="ANNUAL INSURANCE ($)":GOSUB 6600
2240 Q3$="ANNUAL TAXES ($)":GOSUB 6600
2250 Q3$="ANNUAL MAINT ($)":GOSUB 6600
2260 Q3$="MONTHLY INCOME ($)":GOSUB 6600
2270 GOSUB 250:REM CALCULATE PMT
2280 PRINT
2290 P=INT(P*100+0.5)/100
2300 PRINT "MONTHLY PAYMENT ($)":POSITION 24,11:PRINT P
2310 CF=P(7)-P-(P(4)+P(5)+P(6))/12
2320 CF=INT(CF*100+0.5)/100
2330 PRINT "MONTHLY CASH FLOW ($)"
2335 POSITION 24,12:PRINT CF
2340 Q2$="COMPUTE":GOSUB 3400
2350 IF YN$="N" THEN RETURN
2360 GOTO 2170
```

Figure 4.19: Program Listing: Rental Property Analysis

5
DATA ANALYSIS PROGRAMS

This chapter describes a set of data analysis programs that you can use to reduce and plot various kinds of data. We present programs that input data, plot it on the screen, compute moving averages, and perform linear regression analyses. The Plot Data program will also plot data that are reduced by the programs in this chapter.

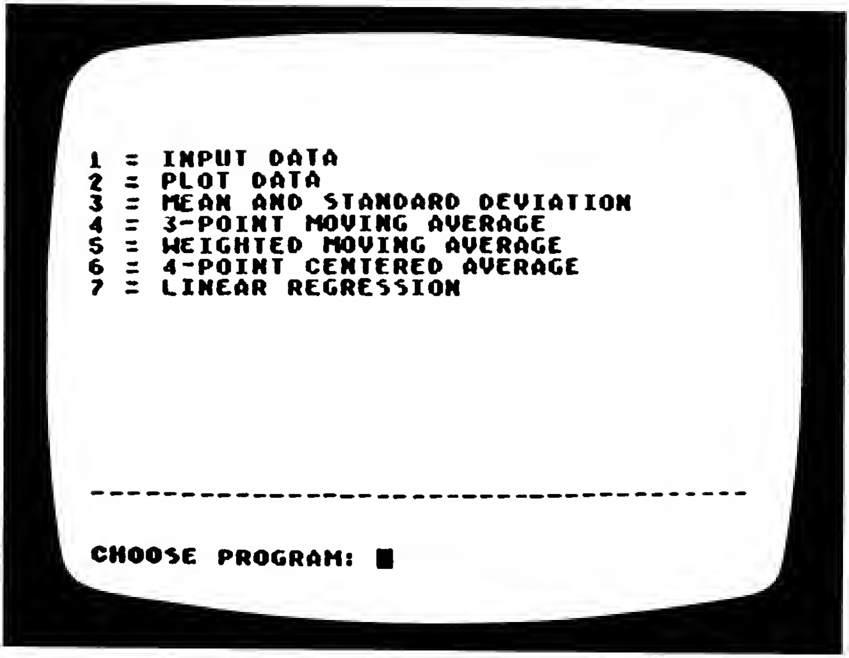
Investors, among others, will find the data analysis programs very useful. You can input daily stock prices and then try the various smoothing programs. These programs smooth out daily price variations and give an indication of longer term trends. The data analysis programs can also be used in many other situations, including analyzing scientific data, plotting and smoothing weight variations, and tracking weather data.

DATA ANALYSIS PROGRAMS MENU

The data analysis programs all work together through common arrays and a common menu program. Notice that the menu lists an input program, a plot program, and several analysis programs. You use the programs together by first entering data with the input program. You can then plot the data or analyze it. Input data are stored in the array D. Data you reduce with the analysis programs are stored in the array R.

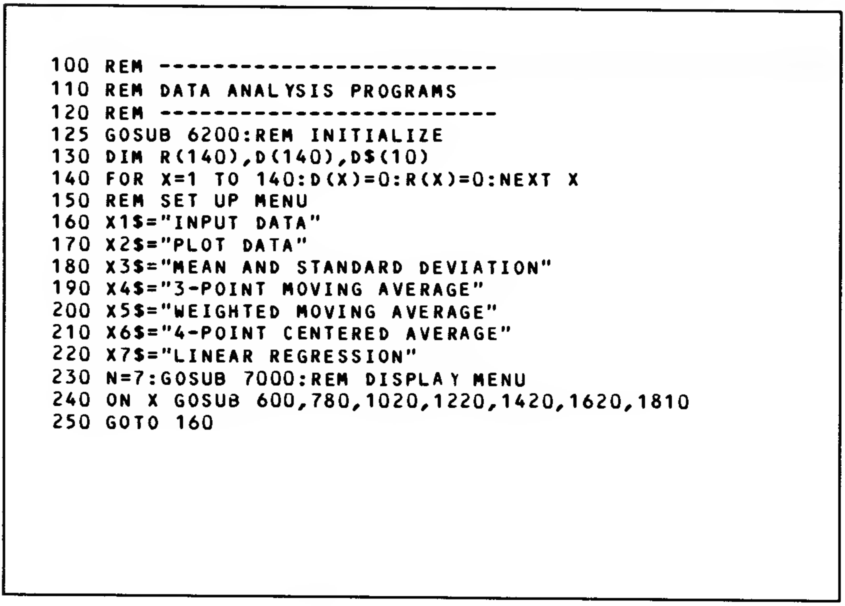
After you run each program you return to the menu. You can then analyze the data again with a different program; you can plot input data, reduced data, or both; or you can enter new data.

Figure 5.1 shows the screen display of the Data Analysis Menu. Figure 5.2 shows the menu program listing.



```
1 = INPUT DATA
2 = PLOT DATA
3 = MEAN AND STANDARD DEVIATION
4 = 3-POINT MOVING AVERAGE
5 = WEIGHTED MOVING AVERAGE
6 = 4-POINT CENTERED AVERAGE
7 = LINEAR REGRESSION




----------------------------------------

CHOOSE PROGRAM: █
```

Figure 5.1: Screen Display: Data Analysis Menu



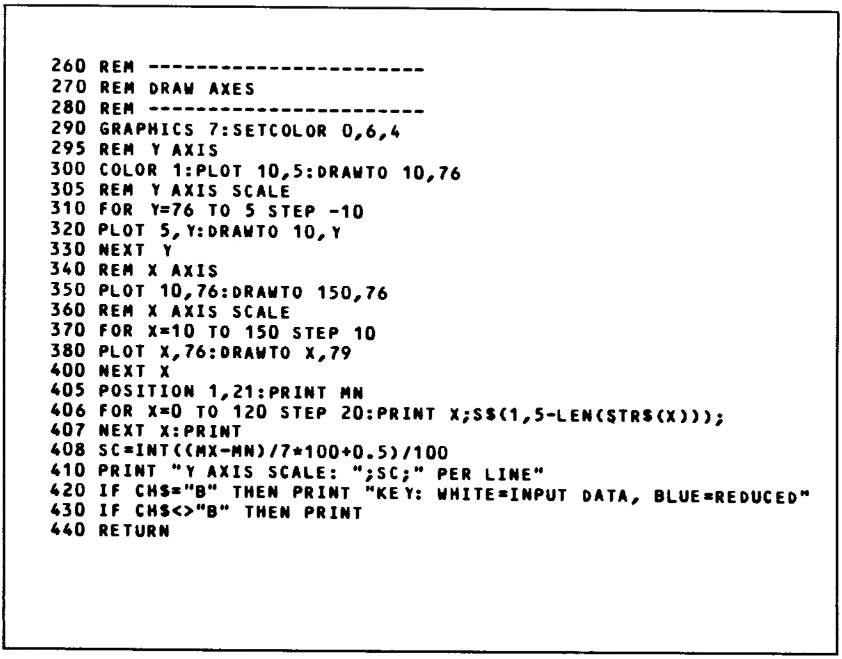
```
100 REM ------------------------
110 REM DATA ANALYSIS PROGRAMS
120 REM ------------------------
125 GOSUB 6200:REM INITIALIZE
130 DIM R(140),D(140),D$(10)
140 FOR X=1 TO 140:D(X)=0:R(X)=0:NEXT X
150 REM SET UP MENU
160 X1$="INPUT DATA"
170 X2$="PLOT DATA"
180 X3$="MEAN AND STANDARD DEVIATION"
190 X4$="3-POINT MOVING AVERAGE"
200 X5$="WEIGHTED MOVING AVERAGE"
210 X6$="4-POINT CENTERED AVERAGE"
220 X7$="LINEAR REGRESSION"
230 N=7:GOSUB 7000:REM DISPLAY MENU
240 ON X GOSUB 600,780,1020,1220,1420,1620,1810
250 GOTO 160
```

Figure 5.2: Program Listing: Data Analysis Menu

DATA ANALYSIS SUBROUTINES

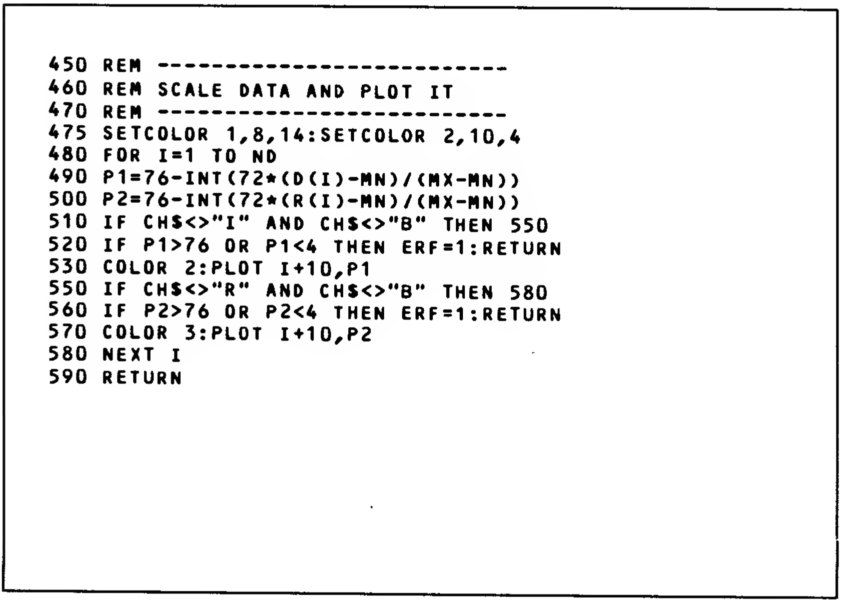
The Plot Data program uses two subroutines to plot data on the screen. One routine draws the axes; the other scales and plots the data. You must enter these programs prior to using the plot program. Figure 5.3 is the program listing for the axis subroutine.

Figure 5.4 shows the program listing for the routine that scales and plots the data. This program plots either input data, reduced data, or both input and reduced data, depending on the user-controlled variable CH$.



```
260 REM ----------------------
270 REM DRAW AXES
280 REM ----------------------
290 GRAPHICS 7:SETCOLOR 0,6,4
295 REM Y AXIS
300 COLOR 1:PLOT 10,5:DRAWTO 10,76
305 REM Y AXIS SCALE
310 FOR Y=76 TO 5 STEP -10
320 PLOT 5,Y:DRAWTO 10,Y
330 NEXT Y
340 REM X AXIS
350 PLOT 10,76:DRAWTO 150,76
360 REM X AXIS SCALE
370 FOR X=10 TO 150 STEP 10
380 PLOT X,76:DRAWTO X,79
400 NEXT X
405 POSITION 1,21:PRINT MN
406 FOR X=0 TO 120 STEP 20:PRINT X;S$(1,5-LEN(STR$(X)));
407 NEXT X:PRINT
408 SC=INT((MX-MN)/7*100+0.5)/100
410 PRINT "Y AXIS SCALE: ";SC;" PER LINE"
420 IF CH$="B" THEN PRINT "KEY: WHITE=INPUT DATA, BLUE=REDUCED"
430 IF CH$<>"B" THEN PRINT
440 RETURN
```

Figure 5.3: Program Listing: Axis Subroutine



```
450 REM ------------------------
460 REM SCALE DATA AND PLOT IT
470 REM ------------------------
475 SETCOLOR 1,8,14:SETCOLOR 2,10,4
480 FOR I=1 TO ND
490 P1=76-INT(72*(D(I)-MN)/(MX-MN))
500 P2=76-INT(72*(R(I)-MN)/(MX-MN))
510 IF CH$<>"I" AND CH$<>"B" THEN 550
520 IF P1>76 OR P1<4 THEN ERF=1:RETURN
530 COLOR 2:PLOT I+10,P1
550 IF CH$<>"R" AND CH$<>"B" THEN 580
560 IF P2>76 OR P2<4 THEN ERF=1:RETURN
570 COLOR 3:PLOT I+10,P2
580 NEXT I
590 RETURN
```

Figure 5.4: Program Listing: Scale and Plot Subroutine

INPUT DATA

Description

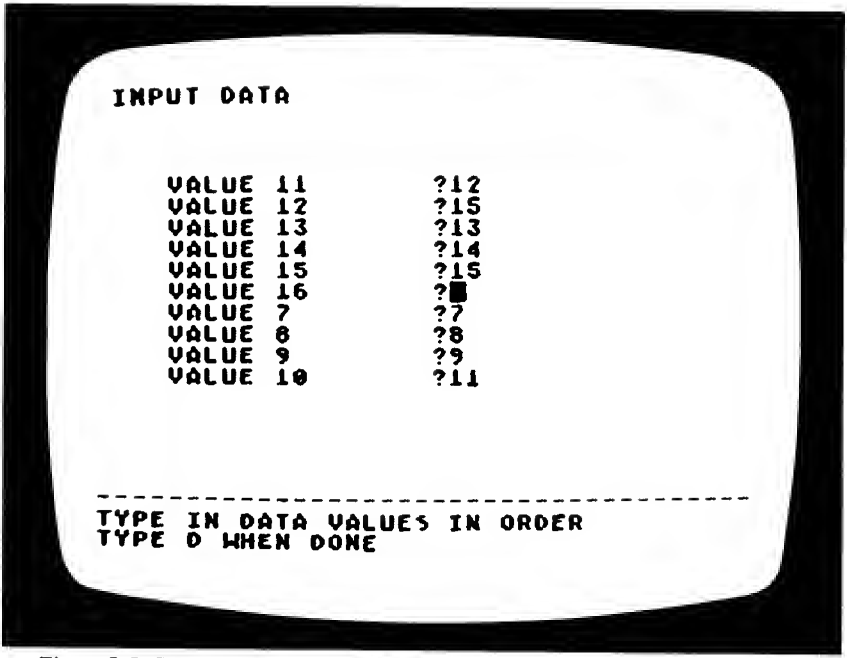
You use this program to enter data that the other programs analyze and plot. You can enter up to 140 data points with the Input Data program. The data are stored in the array D(I). The program prompts you to enter data in the order of their occurrence. If you enter fewer than 140 data points, you type a D to signify that you are done entering data. The input screen is set up to display the last ten points that you enter, so you can check yourself as you go along. Remember, all data that you smooth and plot must first be entered with the Input Data program.

Example

Mary Ann would like to input the data she obtained in a laboratory experiment. The data are as follows:

Point	1	2	3	4	5	6	7	8	9	10
Data	1	3	5	4	4	5	7	8	9	11
Point	11	12	13	14	15	16	17	18	19	20
Data	12	15	13	14	15	16	15	13	12	12
Point	21	22	23	24	25	26	27	28	29	30
Data	11	10	7	9	8	7	5	5	4	3
Point	31	32								
Data	3	0								

Figure 5.5 shows the screen display; Figure 5.6 contains the program listing for the Input Data program.



```
INPUT DATA


    VALUE 11        ?12
    VALUE 12        ?15
    VALUE 13        ?13
    VALUE 14        ?14
    VALUE 15        ?15
    VALUE 16        ?█
    VALUE 7         ?7
    VALUE 8         ?8
    VALUE 9         ?9
    VALUE 10        ?11



---------------------------------------
TYPE IN DATA VALUES IN ORDER
TYPE D WHEN DONE
```

Figure 5.5: Screen Display: Input Data

```
600 REM ------------------------
610 N$="INPUT DATA"
620 REM ------------------------
630 GRAPHICS 0:PRINT N$:PRINT
640 Q1$="TYPE IN DATA VALUES IN ORDER"
650 Q2$="TYPE D WHEN DONE"
660 Q3$="":GOSUB 5000:REM DIALOG
670 ND=0
680 FOR I=1 TO 140
690 J=I-INT((I-1)/10)*10+3
700 POSITION 1,J:PRINT S$:POSITION 5,J
710 PRINT "VALUE ";I:POSITION 20,J
720 INPUT D$
730 IF D$="D" THEN RETURN
740 ND=ND+1
750 D(I)=VAL(D$)
760 NEXT I
770 RETURN
```

Figure 5.6: Program Listing: Input Data

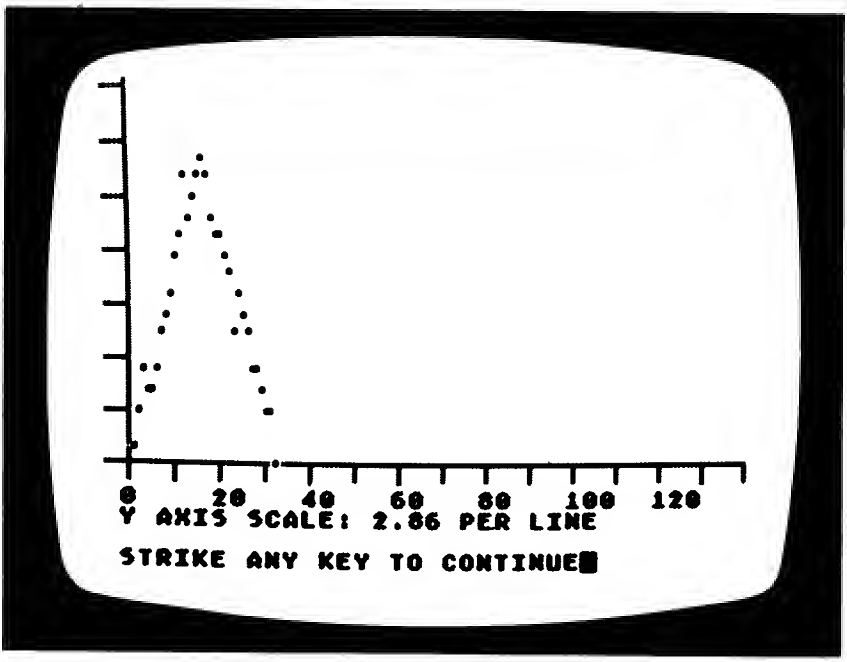
PLOT DATA

Description

The Plot Data program plots either input data or reduced data on the screen. (Reduced data result from using the data analysis programs that follow.) You must tell the program to plot input data, reduced data (data on which you have run one of the data analysis programs), or both by answering the program prompt with an I, R, or B respectively. On a color monitor, white dots represent input data, and blue dots are reduced data. Black-and-white monitors show bright and dim dots. You must also specify the maximum and minimum Y-axis scale values. The screen displays a message showing the data units per Y-axis tick. If your input data exceed your scale values, you will get the message DATA OUT OF RANGE. You must then respecify the scale values.

Example

Let's now plot the input data from the previous example. In this case we choose a maximum value of 20 and a minimum value of 0. We choose to plot input data (I), because we have not reduced it yet. Figure 5.7 shows the screen display for the data plot. Figure 5.8 displays the Plot Data program listing.



```
Y AXIS SCALE: 2.86 PER LINE

STRIKE ANY KEY TO CONTINUE█
```

Figure 5.7: Screen Display: Plot Data

```
780 REM ------------------------
790 N$="PLOT DATA ROUTINE"
800 REM ------------------------
810 Q2$="INPUT OR REDUCED OR BOTH"
820 Q3$="DATA (I OR R OR B)"
830 GOSUB 5000
840 GOSUB 5400:REM GET CHAR
850 IF CH$<>"I" AND CH$<>"R" AND CH$<>"B" THEN 840
860 Q1$="":Q2$=""
870 Q3$="CHOOSE MAX SCALE: "
880 GOSUB 5000:INPUT MX
890 Q3$="CHOOSE MIN SCALE: "
900 GOSUB 5000:INPUT MN
910 GOSUB 260:REM DRAW AXES
920 ERF=0
930 GOSUB 450:REM PLOT DATA
940 IF ERF=0 THEN 980
950 Q1$="DATA OUT OF RANGE"
960 Q2$="PLEASE RESPECIF Y"
970 GOTO 870
980 POSITION 2,24
990 PRINT "STRIKE ANY KEY TO CONTINUE";
1000 GOSUB 5400:REM WAIT FOR CHAR
1010 RETURN
```

Figure 5.8: Program Listing: Plot Data

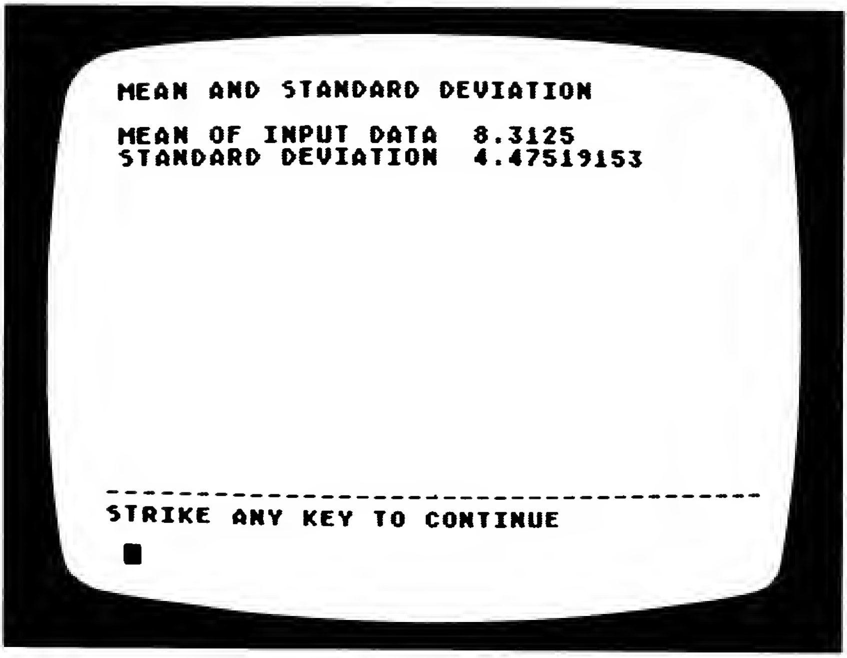
MEAN AND STANDARD DEVIATION

Description

We shall now look at a program that computes the mean and standard deviation for data that have been input with the Input Data program. To use this program you simply select it from the Data Analysis Menu.

Example

Mary Ann would like to compute the mean and standard deviation for the laboratory data she has input. She runs the program and displays the results in Figure 5.9. The program listing for this example appears in Figure 5.10.



```
MEAN AND STANDARD DEVIATION

MEAN OF INPUT DATA  8.3125
STANDARD DEVIATION  4.47519153




-------------------------------------
STRIKE ANY KEY TO CONTINUE
```

Figure 5.9: Screen Display: Mean and Standard Deviation

```
1020 REM -----------------------------
1030 N$="MEAN AND STANDARD DEVIATION"
1040 REM -----------------------------
1050 GRAPHICS 0:PRINT N$:PRINT
1060 REM COMPUTE THE MEAN
1070 ME=0
1080 FOR I=1 TO ND
1090 ME=ME+D(I)
1100 NEXT I
1110 ME=ME/ND
1120 REM COMPUTE STANDARD DEVIATION
1130 SD=0
1140 FOR I=1 TO ND
1150 SD=SD+(D(I)-ME)^2
1160 NEXT I
1170 SD=(SD/ND)^0.5
1180 PRINT "MEAN OF INPUT DATA",ME
1190 PRINT "STANDARD DEVIATION",SD
1200 GOSUB 7400
1210 RETURN
```

Figure 5.10: Program Listing: Mean and Standard Deviation

THREE-POINT MOVING AVERAGE

Description

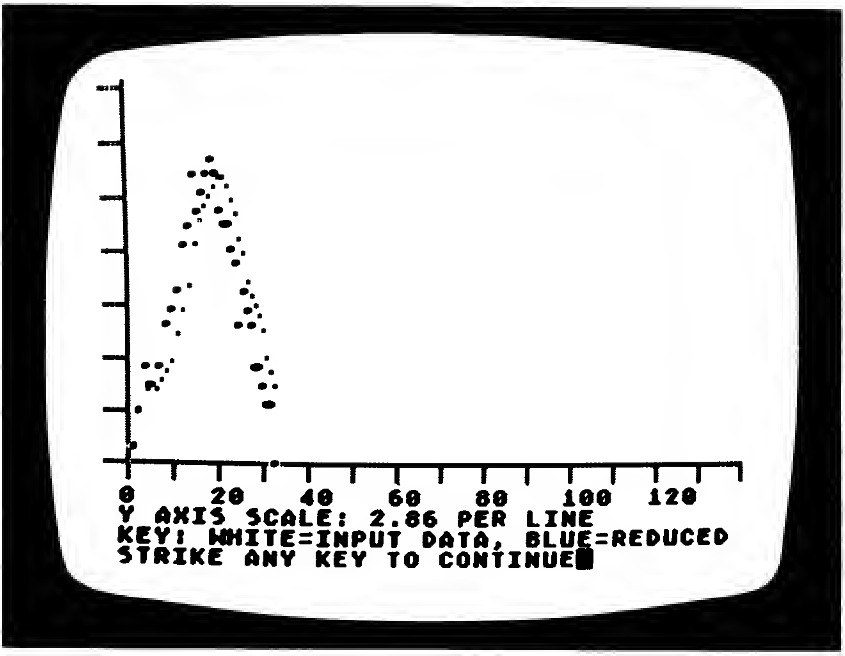
We shall now present three data smoothing programs. The first program computes a three-point moving average on data you store in the Input Data array. The smoothed data are stored in the array R(I). You can display the smoothed data in either numeric or graph form. The three-point moving average takes the average value of the three prior data points and uses the average as the smoothed result. After the program smooths the data, it will ask:

```
DO YOU WANT TO PRINT THE DATA
```

If you answer with a Y, the program displays the data in numeric form. When you return to the main menu, you can graph the smoothed data with the Plot Data program. If you answer with an N, the program returns immediately to the main menu, so that you can run the Plot Data program.

Example

Mary Ann would like to try various smoothing techniques on her laboratory data. She first runs the Three-Point Moving Average program and graphs the results together with the input data. You can see the smoothed data in Figure 5.11; wherever the input data and smoothed data are the same, only the smoothed data are shown. The program listing is shown in Figure 5.12.



```
Y AXIS SCALE: 2.86 PER LINE
KEY: WHITE=INPUT DATA, BLUE=REDUCED
STRIKE ANY KEY TO CONTINUE
```

Figure 5.11: Screen Display: Three-Point Moving Average

```
1220 REM -------------------------
1230 N$="3-POINT MOVING AVERAGE"
1240 REM -------------------------
1250 GRAPHICS 0:PRINT N$:PRINT
1260 REM SET UP THE RESULT IN R(I)
1270 R(1)=0:R(2)=0:R(3)=0
1280 FOR I=4 TO ND
1290 R(I)=(D(I-3)+D(I-2)+D(I-1))/3
1300 NEXT I
1310 Q1$="DO YOU WANT TO PRINT THE"
1320 Q2$="DATA (Y OR N)"
1330 Q3$="":GOSUB 5000
1340 GOSUB 7800:REM ASK YES OR NO
1350 IF YN$="N" THEN RETURN
1360 GRAPHICS 0
1370 FOR I=4 TO ND
1380 PRINT INT(R(I)*100+0.5)/100,
1385 IF INT(I/2)=I/2 THEN PRINT
1390 NEXT I
1400 GOSUB 7400:REM WAIT FOR KEY
1410 RETURN
```

Figure 5.12: Program Listing: Three-Point Moving Average

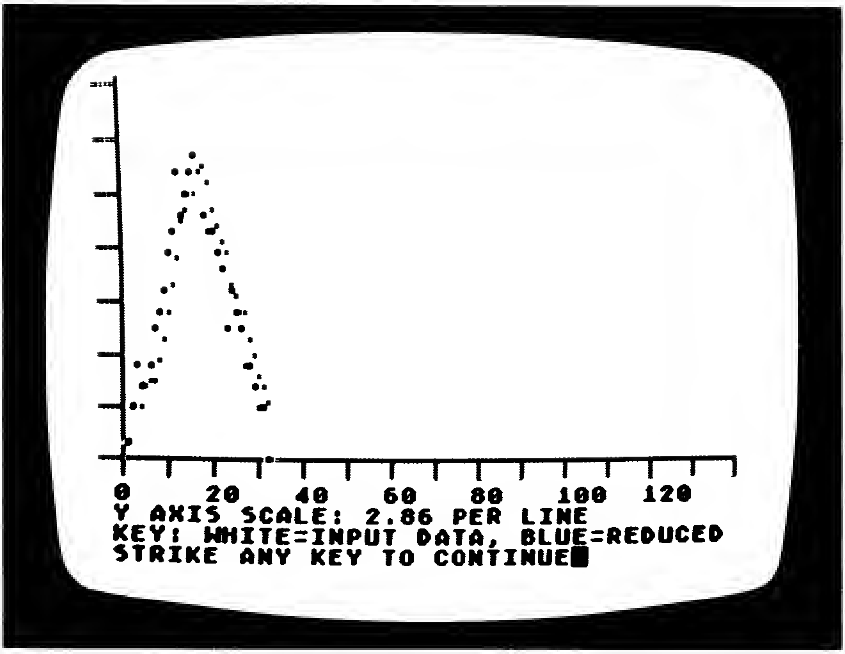
WEIGHTED MOVING AVERAGE

Description

A second smoothing technique is the three-point weighted average. We smooth the data by weighting the most recent point by three, the second most recent point by two, and the third most recent point by one. Thus, the most recent point is given the highest weight. You use this program by calling it from the menu; the resulting data are stored in the array R(I). Again, the data can be displayed numerically or graphed by the Plot Data program. Note that only one set of reduced data can be stored at a time.

Example

Let's use the weighted average program with Mary Ann's laboratory data. In Figure 5.13 we plot both the input data and the reduced data resulting from the Weighted Moving Average program. The program listing is given in Figure 5.14.



```
Y AXIS SCALE: 2.86 PER LINE
KEY: WHITE=INPUT DATA, BLUE=REDUCED
STRIKE ANY KEY TO CONTINUE
```

Figure 5.13: Screen Display: Weighted Moving Average

```
1420 REM -------------------------------
1430 N$="3-PT. WEIGHTED MOVING AVERAGE"
1440 REM -------------------------------
1450 GRAPHICS 0:PRINT N$:PRINT
1460 REM SET UP RESULT IN R(I)
1470 R(1)=0:R(2)=0:R(3)=0
1480 FOR I=4 TO ND
1490 R(I)=(D(I-3)+2*D(I-2)+3*D(I-1))/6
1500 NEXT I
1510 Q1$="DO YOU WANT TO PRINT THE"
1520 Q2$="DATA (Y OR N)"
1530 Q3$="":GOSUB 5000
1540 GOSUB 7800:REM ASK YES OR NO
1550 IF YN$="N" THEN RETURN
1560 GRAPHICS 0
1570 FOR I=4 TO ND
1580 PRINT INT(R(I)*100+0.5)/100,
1585 IF INT(I/2)=I/2 THEN PRINT
1590 NEXT I
1600 GOSUB 7400:REM WAIT FOR KEY
1610 RETURN
```

Figure 5.14: Program Listing: Weighted Moving Average

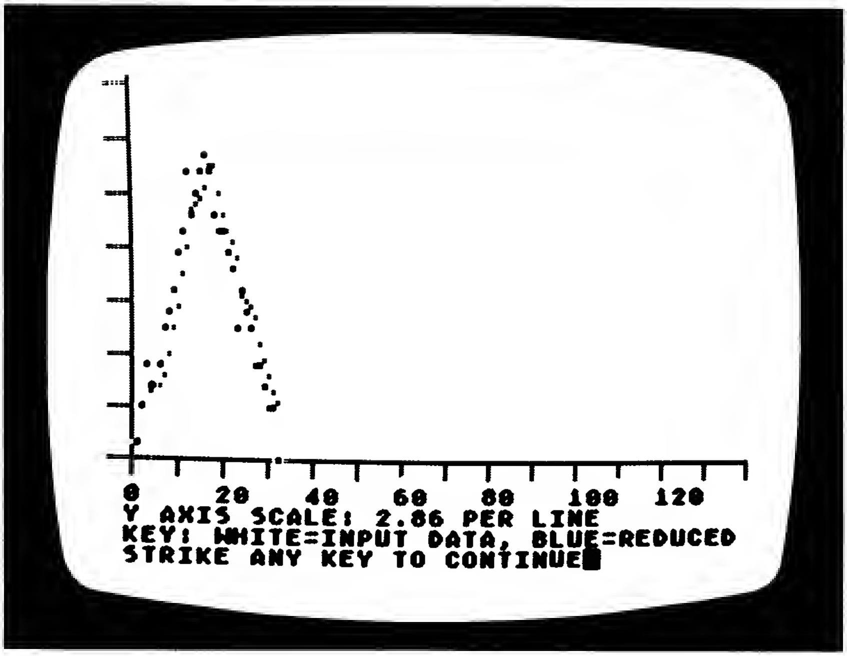
FOUR-POINT CENTERED AVERAGE

Description

We now review a final smoothing technique, the four-point centered average. This technique weights the first and fifth most recent points by a factor of one and the second, third, and fourth most recent points by a factor of two. You use this program by selecting it from the menu. You can list the smoothed data or plot it with the Plot Data program.

Example

Let's apply the centered-average technique to Mary Ann's laboratory data. Figure 5.15 shows the smoothed data along with the input data. Figure 5.16 shows the program listing.



```
Y AXIS SCALE: 2.86 PER LINE
KEY: WHITE=INPUT DATA, BLUE=REDUCED
STRIKE ANY KEY TO CONTINUE
```

Figure 5.15: Screen Display: Four-Point Centered Average

```
1620 REM ------------------------
1630 N$="4-PT. CENTERED AVERAGE"
1640 REM ------------------------
1650 GRAPHICS 0:PRINT N$:PRINT
1660 REM SET UP THE RESULT IN R(I)
1670 FOR I=6 TO ND
1680 R(I)=(D(I-5)+2*(D(I-4)+D(I-3)+D(I-2))+D(I-1))/8
1690 NEXT I
1700 Q1$="DO YOU WANT TO PRINT THE"
1710 Q2$="DATA (Y OR N)"
1720 Q3$="":GOSUB 5000
1730 GOSUB 7800:REM ASK YES OR NO
1740 IF YN$="N" THEN RETURN
1750 GRAPHICS 0
1760 FOR I=4 TO ND
1770 PRINT INT(R(I)*100+0.5)/100,
1775 IF INT(I/2)=I/2 THEN PRINT
1780 NEXT I
1790 GOSUB 7400:REM WAIT FOR KEY
1800 RETURN
```

Figure 5.16: Program Listing: Four-Point Centered Average

LINEAR REGRESSION

Description

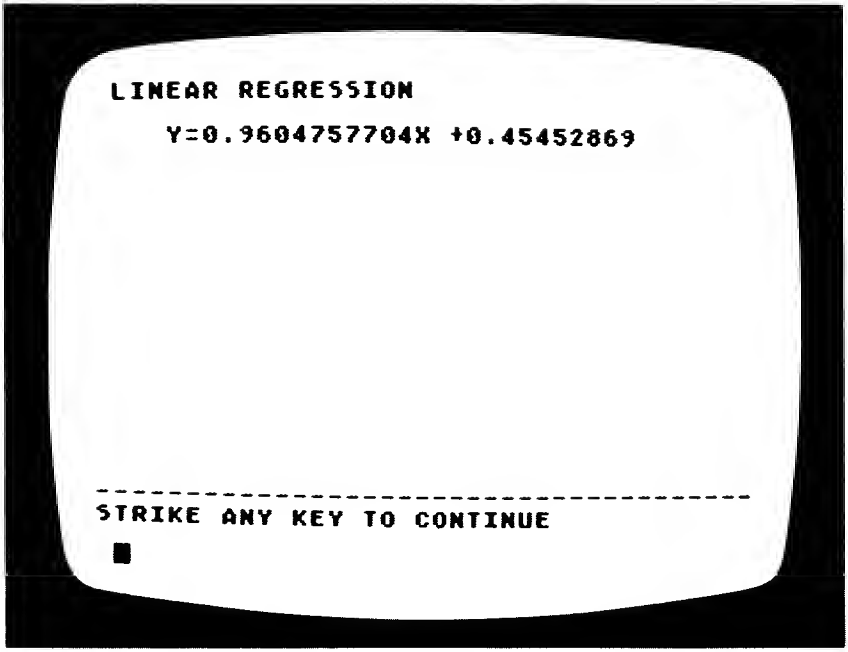
The linear regression technique allows you to find the "best" straight line for a set of data. The selected line is the one that best meets the regression criteria. You run the Linear Regression program by calling it from the program menu. It uses the data stored with the Input Data program. The program displays the equation for the straight line on your screen. You can also plot the straight line with the plot program, but the resulting plot will not appear straight because of the vertical resolution of the display.

Example

Frank has obtained data from an experiment and would like the equation for the best straight line for these data. The data are:

Point	1	2	3	4	5	6	7	8	9	10	11
Data	1	2	3	5	5	6	7	9	10	11	11
Point	12	13	14	15	16	17	18	19	20	21	22
Data	12	13	14	14	15	16	17	19	20	21	22

The screen display for this example appears in Figure 5.17. The program listing is shown in Figure 5.18.



```
LINEAR REGRESSION
    Y=0.9604757704X +0.45452869




----------------------------------------
STRIKE ANY KEY TO CONTINUE
 ■
```

Figure 5.17: Screen Display: Linear Regression

```
1810 REM ------------------------
1820 N$="LINEAR REGRESSION"
1830 REM ------------------------
1840 GRAPHICS 0:PRINT N$:PRINT
1850 SX=0:SY=0:SXY=0:SX2=0
1860 FOR I=1 TO ND
1870 SX=SX+I
1880 SY=SY+D(I)
1890 SXY=SXY+D(I)*I
1900 SX2=SX2+I^2
1910 NEXT I
1920 M=(SXY-SX*SY/ND)/(SX2-SX^2/ND)
1930 B=SY/ND-M*SX/ND
1940 PRINT "    Y=";M;"X +";B
1950 REM UPDATE RESULT ARRAY
1960 FOR I=1 TO ND
1970 R(I)=B+M*I
1980 NEXT I
1990 GOSUB 7400
2000 RETURN
```

Figure 5.18: Program Listing: Linear Regression

6 RECORD KEEPING PROGRAMS

This chapter shows you how to use the diskette storage system to store, retrieve, and analyze data. We shall first create a personal file that you can use to store names and addresses. You can use the personal files you create for phone directories, business mailing lists, and other applications. Then we'll show you how to create an automobile file that will make it easy to keep mileage records for your car.

Disk file programs usually don't appear in collections of programs, because they can be very complicated. But by using our central subroutine library, we have been able to reduce the usual complexity of these disk file programs. The results are programs that are easy to enter and use.

Prior to using the programs in this chapter, you must add several new subroutines to the SUBLIB file. These subroutines begin at line numbers 3000, 3800, and 4600; the subroutine listings are in Appendix A.

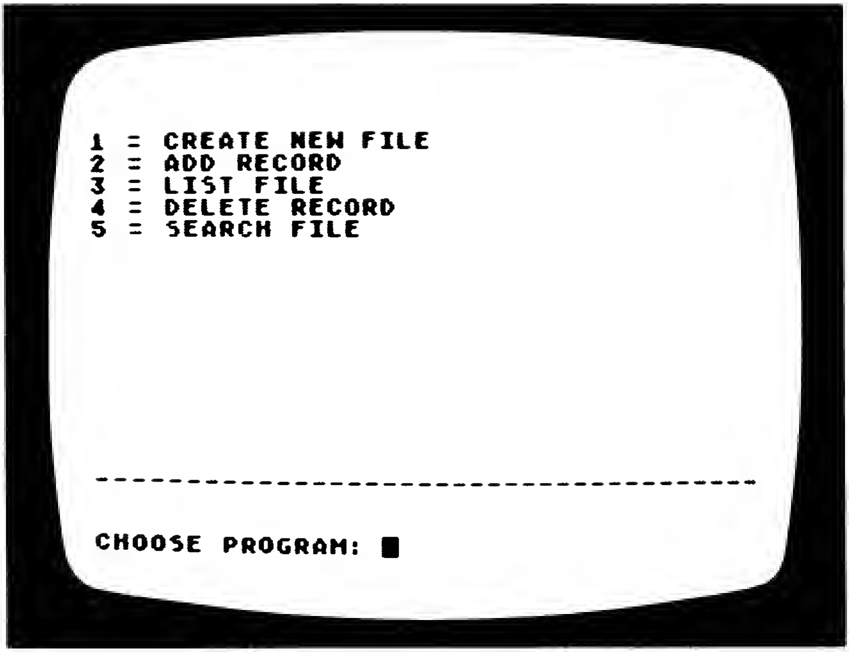
Unlike the other chapters, the programs in this chapter must be saved in *two* files—one containing the Personal File programs and the other containing the Automobile File programs. Each set of programs begins with a menu.

These programs automatically create data files on disk drive A. You can add these files to your program disk, or you can use a freshly formatted disk.

First let's define a few terms. An *item* is the smallest quantity stored in a file. For the personal file the items are last name, first name, telephone number, street address, city, state, and zip code. The items in the auto record are date, odometer reading, and gallons of gas used. A *record* is defined as a set of items. For the personal file a record is the name, address, and phone number for each individual. Finally, a *file* is a set of records.

PERSONAL FILE MENU

The programs for the personal file run from a menu. As you can see, the individual programs allow you to create a file, add records, delete records, list the entire file, and search for individual records in the file. When you begin, you will create your file using the Create File program. You then add records to your file with the Add Record program. The Delete Record program removes records from your file; you also change a record by deleting it and then reentering it with new information. The List File program displays the entire file on your screen or printer. The Search Record program allows you to find all records with the same last name. Figure 6.1 shows the menu screen display. Figure 6.2 shows the menu program listing.



```
1 = CREATE NEW FILE
2 = ADD RECORD
3 = LIST FILE
4 = DELETE RECORD
5 = SEARCH FILE




---------------------------------------

CHOOSE PROGRAM: █
```

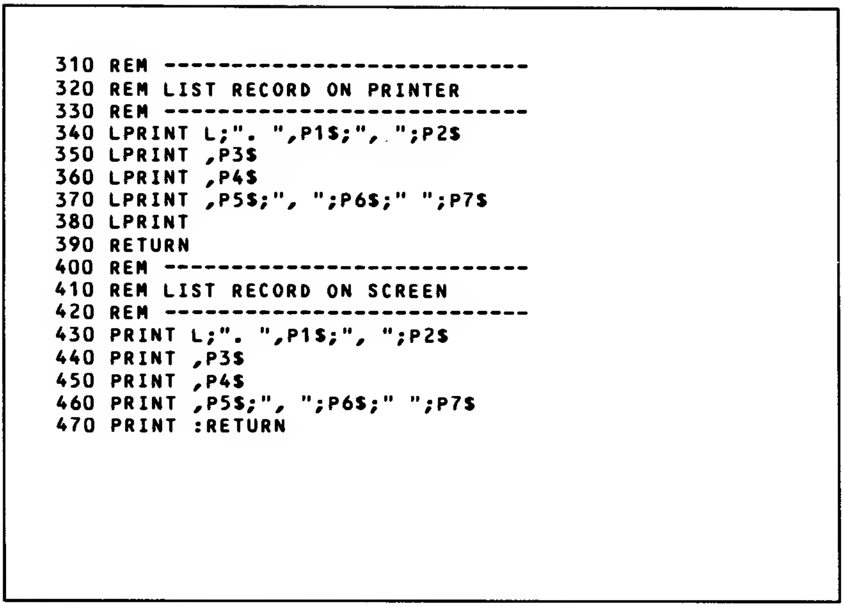
Figure 6.1: Screen Display: Personal File Menu

```
100 REM ----------------------------
110 REM PERSONAL FILE
120 REM ----------------------------
130 GOSUB 6200:REM INITIALIZE
140 X1$="CREATE NEW FILE"
150 X2$="ADD RECORD"
160 X3$="LIST FILE"
170 X4$="DELETE RECORD"
180 X5$="SEARCH FILE"
190 REM DECLARE NO. OF FIELDS AND NAMES
200 F1$="LAST NAME"
210 F2$="FIRST NAME"
220 F3$="TELEPHONE"
230 F4$="STREET ADD"
240 F5$="CITY"
250 F6$="STATE"
260 F7$="ZIP"
270 NF=7:F$="D:PERS.DAT":FB$="D:PERSBAK.DAT"
280 N=5:GOSUB 7000:REM DISPLAY MENU
290 ON X GOSUB 480,680,740,960,1050
300 GOTO 140
```

Figure 6.2: Program Listing: Personal File Menu

PERSONAL FILE SUBROUTINES

As you enter the menu program you should also enter the two subroutines shown in Figure 6.3. These routines are used by the List Personal File program to list your file on the screen or on your printer. They handle the formatting differences between the two output devices. The screen program also contains a delay function, which scrolls the screen slowly enough to read the names as they go by.



```
310 REM ---------------------------
320 REM LIST RECORD ON PRINTER
330 REM ---------------------------
340 LPRINT L;". ",P1$;", ";P2$
350 LPRINT ,P3$
360 LPRINT ,P4$
370 LPRINT ,P5$;", ";P6$;" ";P7$
380 LPRINT
390 RETURN
400 REM ---------------------------
410 REM LIST RECORD ON SCREEN
420 REM ---------------------------
430 PRINT L;". ",P1$;", ";P2$
440 PRINT ,P3$
450 PRINT ,P4$
460 PRINT ,P5$;", ";P6$;" ";P7$
470 PRINT :RETURN
```

Figure 6.3: Program Listing: List Personal File Subroutines

CREATE FILE AND ADD RECORD

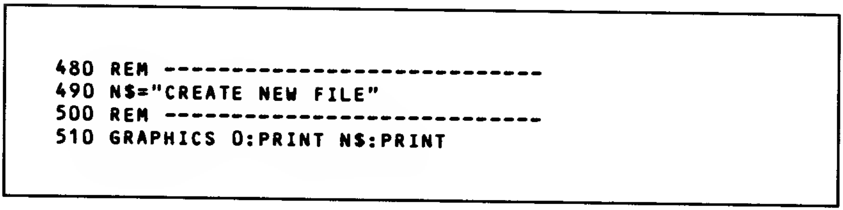
Description

The Create File program is used when you first set up your personal file. This program initializes the disk and disk directory and prepares the disk for data entry. If you use Create File on a disk already containing a personal file, the previous file will be erased. (The program warns you that the file is going to be erased, and it asks your permission to proceed.) Select the Create File program by displaying the Personal File Menu and pressing key 1.

The Add Record program is the one you use to add information to your file. You run the program by displaying the Personal File Menu and depressing the 2 on your keyboard. As with the Create File program, you are asked to enter the information that will appear in your file: last name, first name, telephone number, street address, city, state, and zip code. Records are stored in the order in which they are added.

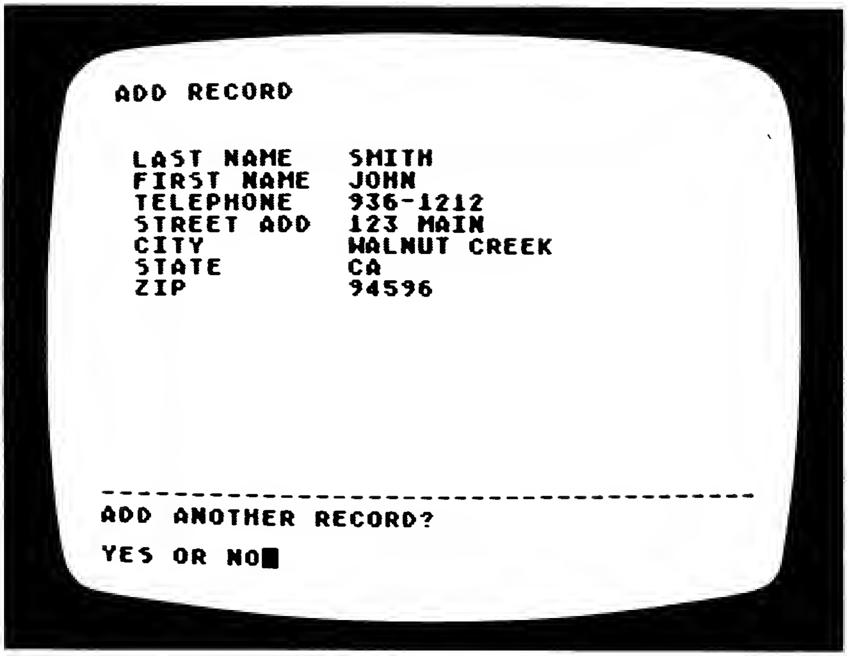
Example

Figure 6.4 shows a screen display for a typical Add Record dialog. The Create File program had to be used first to create the file. Figure 6.5 shows the listings for both the Create File and Add Record programs.



```
480 REM --------------------------
490 N$="CREATE NEW FILE"
500 REM --------------------------
510 GRAPHICS 0:PRINT N$:PRINT
```

Figure 6.5: Program Listing: Create File and Add Record (continues)



```
ADD RECORD

  LAST NAME    SMITH
  FIRST NAME   JOHN
  TELEPHONE    936-1212
  STREET ADD   123 MAIN
  CITY         WALNUT CREEK
  STATE        CA
  ZIP          94596

--------------------------------------
ADD ANOTHER RECORD?

YES OR NO
```

Figure 6.4: Screen Display: Add Record

```
520 K=1
530 GOSUB 5800:GOSUB 5800:GOSUB 5800:GOSUB 5800
540 GOSUB 5800:GOSUB 5800:GOSUB 5800
560 REM OPEN UP NEW FILE AND WRITE REC
570 OPEN #1,8,0,F$
580 PRINT #1;P1$;EL$;P2$;EL$;P3$;EL$;P4$
590 PRINT #1;P5$;EL$;P6$;EL$;P7$;EL$;P8$
600 CLOSE #1
605 GRAPHICS 0:PRINT N$:PRINT
610 POSITION 2,5
620 PRINT "NEW FILE IS CREATED."
630 PRINT
640 PRINT "FROM NOW ON, USE ADD RECORD"
650 PRINT "TO ADD TO IT."
660 GOSUB 7400:REM PAUSE
670 RETURN
680 REM -------------------------
690 N$="ADD RECORD"
700 REM -------------------------
710 REM CALL ADD RECORD SUBROUTINE
720 GOSUB 3000
730 RETURN
```

Figure 6.5: Program Listing: Create File and Add Record

LIST PERSONAL FILE

Description

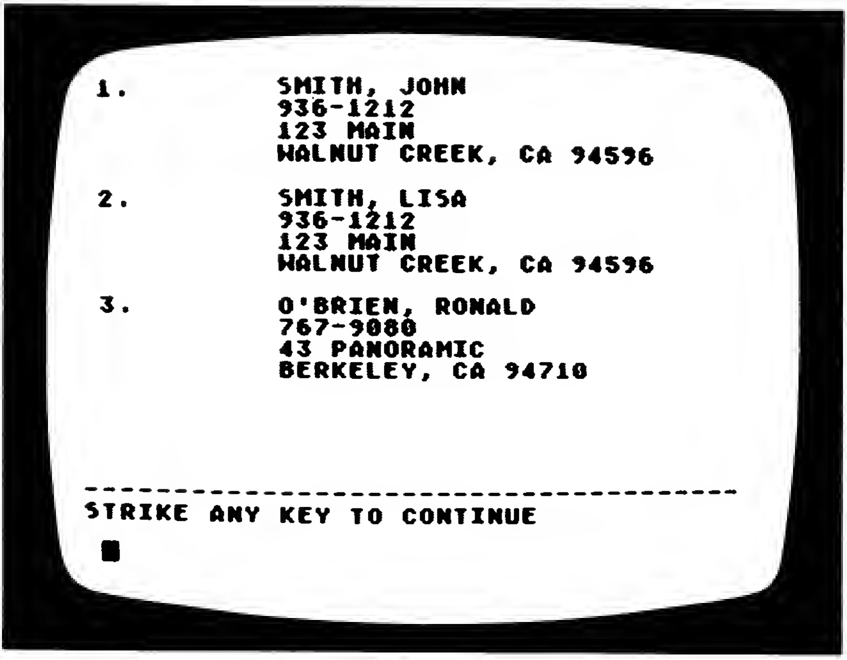
This program lists your personal file. You run the program by activating the Personal File Menu program and depressing the 3 on your keyboard. The program asks the following question:

```
OUTPUT TO PRINTER
YES OR NO
```

Striking a Y causes the entire personal file to be listed on your printer. Striking an N causes the file to be on your screen. Records will scroll on the screen; a delay function scrolls the screen slowly enough for you to read the records.

Example

Figure 6.6 shows a screen display of a personal file. Note that each record has a record number and that blank lines separate individual records. You will use the record number when you delete records from the file. Figure 6.7 displays the program listing for the List Personal File program.



```
1.          SMITH, JOHN
            936-1212
            123 MAIN
            WALNUT CREEK, CA 94596

2.          SMITH, LISA
            936-1212
            123 MAIN
            WALNUT CREEK, CA 94596

3.          O'BRIEN, RONALD
            767-9080
            43 PANORAMIC
            BERKELEY, CA 94710



---------------------------------------
STRIKE ANY KEY TO CONTINUE
 ■
```

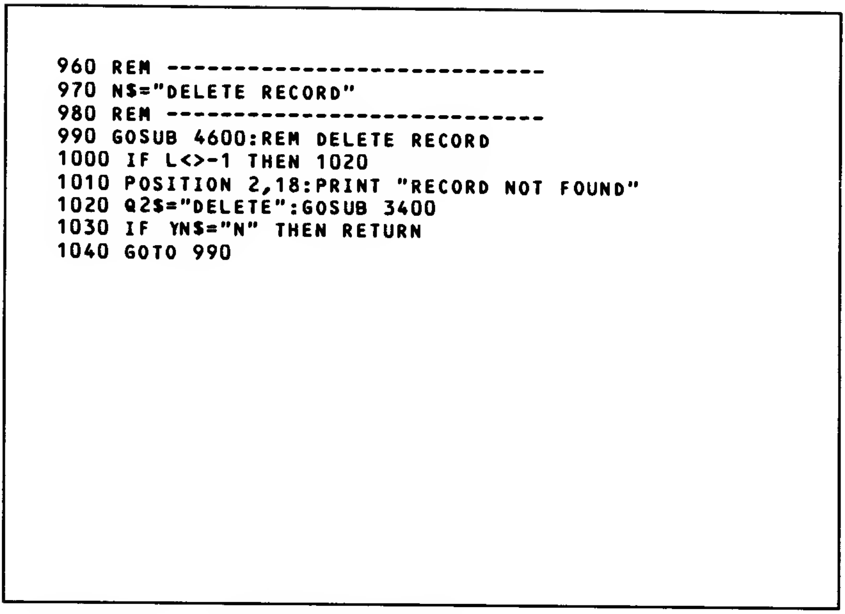
Figure 6.6: Screen Display: List Personal File

```
740 REM ---------------------------
750 N$="LIST FILE"
760 REM ---------------------------
770 GRAPHICS 0:PRINT N$:PRINT
780 Q1$="OUTPUT TO PRINTER"
790 Q2$="":Q3$="YES OR NO"
800 GOSUB 5000:REM DIALOG
810 GOSUB 7800:REM YESNO
820 GRAPHICS 0
830 L=1
840 TRAP 920
850 OPEN #1,4,0,F$
860 INPUT #1;P1$,P2$,P3$,P4$,P5$,P6$,P7$,P8$
870 IF YN$="Y" THEN GOSUB 310:GOTO 910
880 GOSUB 400
890 FOR I1=1 TO 150:REM DELA Y TO READ ON SCREEN
900 NEXT I1
910 L=L+1:GOTO 860
920 CLOSE #1
930 PRINT :PRINT :PRINT
940 GOSUB 7400:REM PAUSE
950 RETURN
```

Figure 6.7: Program Listing: List Personal File

DELETE RECORD

This program is used to delete old information from your personal file. You can correct information by first deleting it with the Delete Record program and then reentering correct data with the Add Record program. You use this program by indicating the number of the record you wish to delete. All subsequent records are renumbered so that the numbering remains consecutive. You can obtain the record number from the List Personal File program or the Search Personal File program. Figure 6.8 is the Delete Record program listing.



```
960 REM ----------------------------
970 N$="DELETE RECORD"
980 REM ----------------------------
990 GOSUB 4600:REM DELETE RECORD
1000 IF L<>-1 THEN 1020
1010 POSITION 2,18:PRINT "RECORD NOT FOUND"
1020 Q2$="DELETE":GOSUB 3400
1030 IF YN$="N" THEN RETURN
1040 GOTO 990
```

Figure 6.8: Program Listing: Delete Record

SEARCH PERSONAL FILE

Description

The Search Personal File program allows you to find an individual record without listing the entire file. You enter the last name of the individual you are searching for. The first record having the same last name you specify is displayed on your screen. The program then asks if you would like to search for additional matches. The program tells you if no last name matches are found.

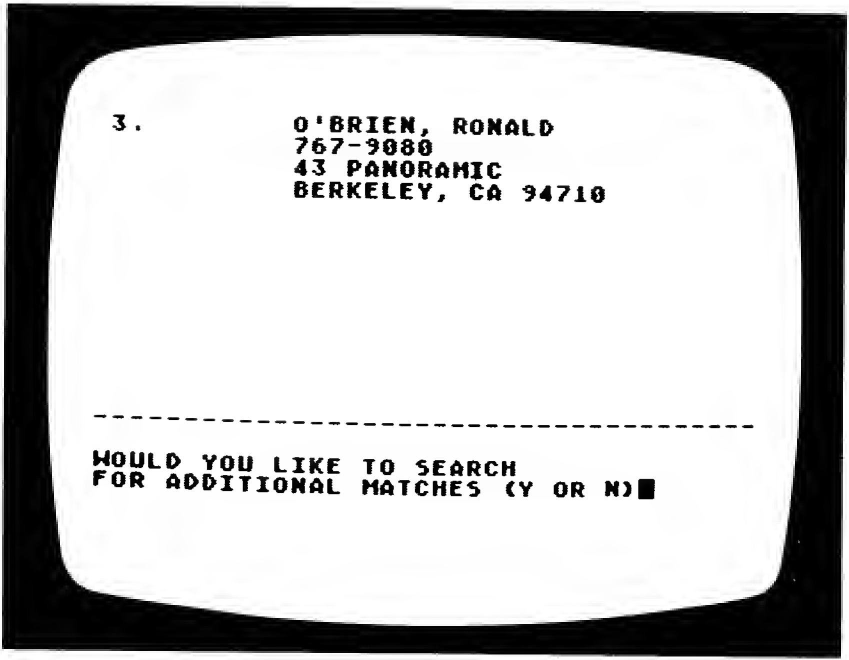
The spelling you specify must match exactly the spelling in your file. Therefore, you must be consistent in the way you add records to your file. If you enter DU BOIS as the last name in your record and you search for the name DUBOIS, the program will tell you that no match is found.

Example

Let's search for the record for RONALD O'BRIEN. The screen display is shown in Figure 6.9. Figure 6.10 is the Search Personal File program listing.

```
1050 REM --------------------------
1060 N$="SEARCH FILE"
1070 REM --------------------------
1080 GRAPHICS 0:PRINT N$:PRINT :F=0:L=0
```

Figure 6.10: Program Listing: Search Personal File (continues)



```
3.          O'BRIEN, RONALD
            767-9080
            43 PANORAMIC
            BERKELEY, CA 94710




---------------------------------------

WOULD YOU LIKE TO SEARCH
FOR ADDITIONAL MATCHES (Y OR N)
```

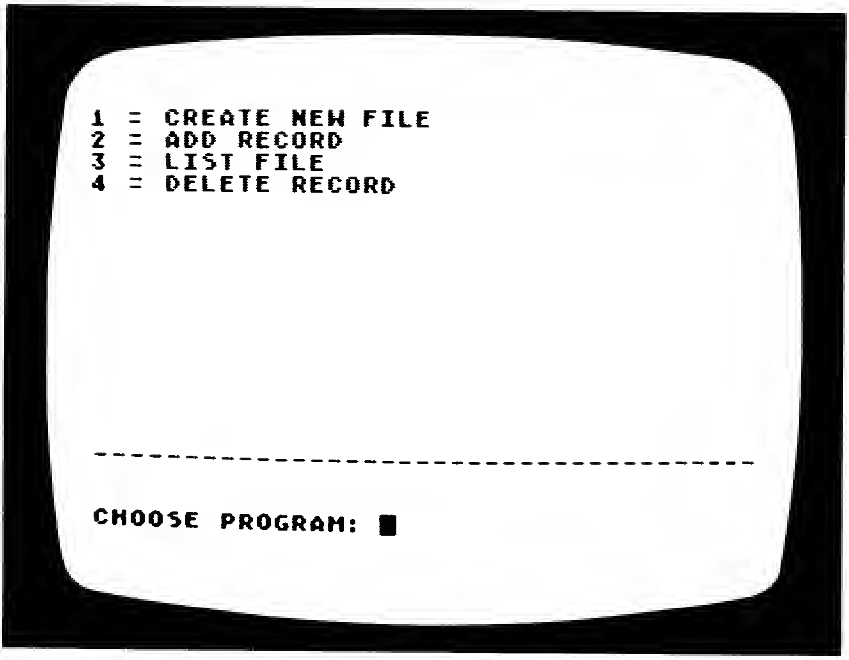
Figure 6.9: Screen Display: Search Personal File

```
1090 Q1$="SEARCH FOR LAST NAME"
1100 Q2$="":Q3$="LAST NAME: "
1110 GOSUB 5000
1120 INPUT X9$
1130 TRAP 1220
1140 OPEN #1,4,0,F$
1150 L=L+1
1160 INPUT #1;P1$,P2$,P3$,P4$,P5$,P6$,P7$,P8$
1170 IF P1$<>X9$ THEN 1150
1180 F=1:REM SET FOUND FLAG
1190 GRAPHICS 0:POSITION 2,5:GOSUB 400:REM DISPLAY REC
1200 Q1$="":Q2$="WOULD YOU LIKE TO SEARCH"
1202 Q3$="FOR ADDITIONAL MATCHES (Y OR N)"
1204 GOSUB 5000:GOSUB 7800:REM YES-NO
1206 IF YN$="N" THEN 1220
1210 GOTO 1150:REM LOOK FOR NEXT MATCH
1220 CLOSE #1:REM OUT OF RECORDS
1230 POSITION 2,18
1240 IF F=0 THEN PRINT "NO MATCHES FOUND"
1250 Q2$="SEARCH":GOSUB 3400
1260 IF YN$="N" THEN RETURN
1270 GOTO 1080
```

Figure 6.10: Program Listing: Search Personal File

AUTOMOBILE FILE MENU

We will now develop a disk file that keeps mileage records for your automobile. You can use these programs to enter a calendar date, odometer reading, and amount of gasoline purchased. The List Automobile File program computes the average miles per gallon and returns that figure in the listing. You call the Automobile Record programs from a menu program. Figure 6.11 shows the screen display for the Automobile Record menu. Figure 6.12 is the menu program listing.



```
1 = CREATE NEW FILE
2 = ADD RECORD
3 = LIST FILE
4 = DELETE RECORD




--------------------------------------

CHOOSE PROGRAM: █
```

Figure 6.11: Screen Display: Automobile Record Menu

```
100 REM ----------------------------
110 REM AUTOMOBILE FILE
120 REM ----------------------------
130 GOSUB 6200:REM INITIALIZE
140 X1$="CREATE NEW FILE"
150 X2$="ADD RECORD"
160 X3$="LIST FILE"
170 X4$="DELETE RECORD"
180 REM DECLARE NO. OF FIELDS AND NAMES
190 F1$="DATE"
200 F2$="ODOMETER"
210 F3$="GALLONS"
220 NF=3:F$="D:CAR.DAT":FB$="D:CARBAK.DAT"
230 N=4:GOSUB 7000:REM DISPLAY MENU
240 ON X GOSUB 260,450,510,770
250 GOTO 230
```

Figure 6.12: Program Listing: Automobile Record Menu

CREATE FILE AND ADD RECORD

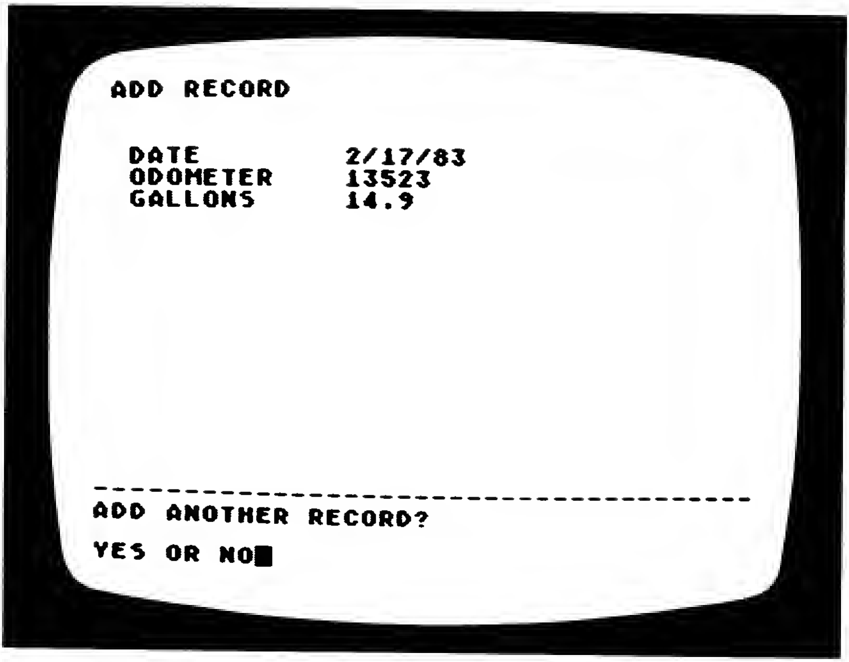
Description

The Create File program is used when you first set up your automobile file. This program initializes the disk and disk directory and prepares the disk for data entry. Run this program only once. If you want to keep records on more than one car, you must use different diskettes.

You use the Add Record program to add records to your automobile file. You follow the format established in the Create File program by entering the date, odometer reading, and gallons of gasoline purchased. Again, records are stored in the order in which they are added. If you make an error as you create a record, the average miles per gallon may be incorrectly computed. You should check your records carefully immediately after you enter them, because an incorrect record is difficult to correct once you have added other records.

Example

On February 17, 1982, you purchase 14.9 gallons of gas with an odometer reading of 13,523 miles. Figure 6.13 shows the screen display as you add this information to the file you previously created. Figure 6.14 is the program listing for the Create File and Add Record programs.



```
ADD RECORD

 DATE          2/17/83
 ODOMETER      13523
 GALLONS       14.9








--------------------------------------
ADD ANOTHER RECORD?

YES OR NO█
```

Figure 6.13: Screen Display: Add Record

```
260 REM -----------------------------
270 N$="CREATE NEW FILE"
280 REM -----------------------------
290 GRAPHICS 0:PRINT N$:PRINT
300 K=1
310 GOSUB 5800:GOSUB 5800:GOSUB 5800
320 REM OPEN UP NEW FILE AND WRITE REC
330 OPEN #1,8,0,F$
340 PRINT #1;P1$;EL$;P2$;EL$;P3$;EL$;P4$
350 PRINT #1;P5$;EL$;P6$;EL$;P7$;EL$;P8$
360 CLOSE #1
370 GRAPHICS 0:PRINT N$:PRINT
380 POSITION 2,5
390 PRINT "NEW FILE IS CREATED."
400 PRINT
410 PRINT "FROM NOW ON, USE ADD RECORD"
420 PRINT "TO ADD TO IT."
430 GOSUB 7400:REM PAUSE
440 RETURN
450 REM -----------------------------
460 N$="ADD RECORD"
470 REM ---------------------------
480 REM CALL ADD RECORD SUBROUTINE
490 GOSUB 3000
500 RETURN
```

Figure 6.14: Program Listing: Create File and Add Record

LIST AUTOMOBILE FILE

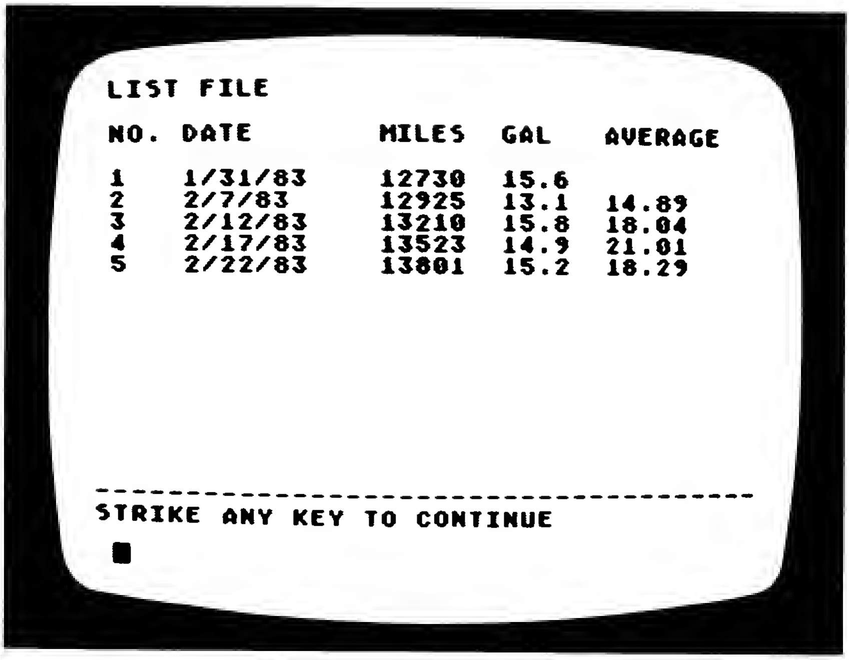
Description

The List Automobile File program lists the contents of your automobile record on your screen. This program also computes and displays the average miles per gallon for each record. The screen will scroll slowly enough for you to read the information.

You can easily monitor your car's performance with this program. A sudden drop in average miles per gallon may indicate a problem with your car.

Example

You would like to view the contents of your automobile log. Figure 6.15 is the screen display for this example. Figure 6.16 is the List Automobile File program listing.



```
LIST FILE

NO. DATE        MILES  GAL   AVERAGE

1   1/31/83     12730  15.6
2   2/7/83      12925  13.1  14.89
3   2/12/83     13210  15.8  18.04
4   2/17/83     13523  14.9  21.01
5   2/22/83     13801  15.2  18.29




----------------------------------------
STRIKE ANY KEY TO CONTINUE
 ■
```

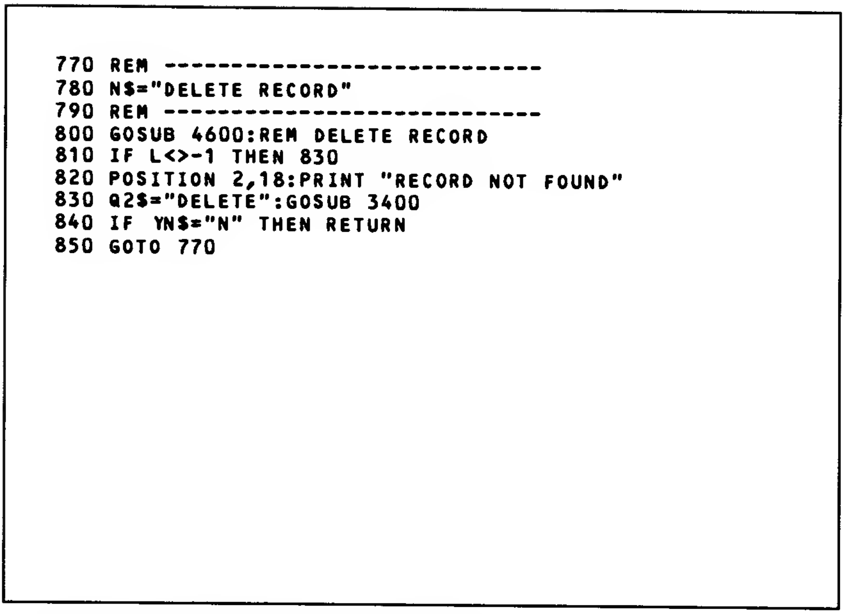
Figure 6.15: Screen Display: List Automobile File

```
510 REM ----------------------------
520 N$="LIST FILE"
530 REM ----------------------------
540 GRAPHICS 0:PRINT N$:PRINT
550 MI=0:GAL=0
560 PRINT "NO. DATE       MILES  GAL   AVERAGE"
570 L=1
580 TRAP 740
590 OPEN #1,4,0,F$
600 INPUT #1;P1$,P2$,P3$,P4$,P5$,P6$,P7$,P8$
610 GAL=VAL(P3$)
620 AVE=(VAL(P2$)-MI)/GAL
630 MI=VAL(P2$)
640 POSITION 2,L+3:PRINT L
650 POSITION 6,L+3:PRINT P1$
660 POSITION 17,L+3:PRINT P2$
670 POSITION 24,L+3:PRINT P3$
680 IF L=1 THEN 700
690 POSITION 30,L+3:PRINT INT(100*AVE+0.5)/100
700 FOR I1=1 TO 100:REM DELAY TO READ ON SCREEN
710 NEXT I1
720 L=L+1
730 GOTO 600
740 CLOSE #1
750 GOSUB 7400:REM PAUSE
760 RETURN
```

Figure 6.16: Program Listing: List Automobile File

DELETE RECORD

The Delete Record program allows you to delete a record from your file. You can use this program to delete information that you enter incorrectly. You delete a record by specifying its number. When you delete a record in the middle of a file, all subsequent records are renumbered so that there are no holes in the file. Figure 6.17 shows the Delete Record program listing. You must delete records immediately upon incorrect entry; otherwise, the program will incorrectly compute miles per gallon.



```
770 REM ----------------------------
780 N$="DELETE RECORD"
790 REM ----------------------------
800 GOSUB 4600:REM DELETE RECORD
810 IF L<>-1 THEN 830
820 POSITION 2,18:PRINT "RECORD NOT FOUND"
830 Q2$="DELETE":GOSUB 3400
840 IF YN$="N" THEN RETURN
850 GOTO 770
```

Figure 6.17: Program Listing: Delete Record

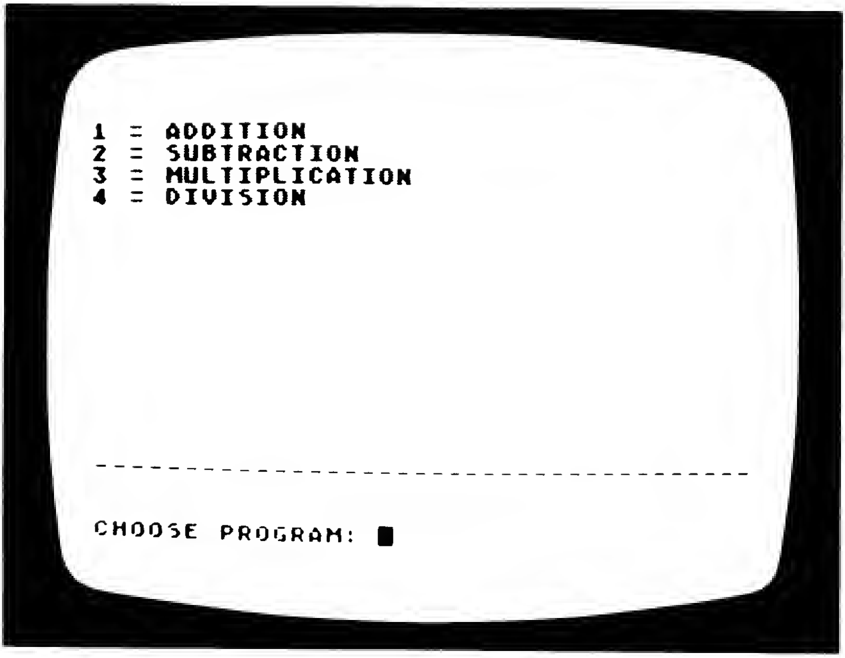
7
MATHEMATICS PRACTICE PROGRAMS

This chapter presents three sets of programs that allow you to use your computer to practice arithmetic. You must save each set in its own file. Each set runs from a menu, which you enter first. We provide drills in addition, subtraction, multiplication, and division, with both whole numbers and fractions. The three menus provide programs at multiple levels of difficulty; the programs are suitable for students from elementary to high school level. Each program keeps track of right and wrong answers. You will be greeted with a sound when you answer incorrectly. You will find that these programs can greatly sharpen your math skills; they are also helpful in completing homework assignments!

MATHEMATICS PRACTICE I MENU

The first set of four programs provides practice in adding, subtracting, multiplying, and dividing two whole numbers. You begin this set of programs by displaying the Mathematics Practice I Menu. Depress a number key to choose the drill you want. Once you have chosen the drill, the program asks you to choose the largest number you want to work with. Then the program presents you with two numbers and an operator to indicate addition, subtraction, multiplication, or division. You enter your answer from left to right. The programs generate a beep when you answer incorrectly. Each program keeps track of the total number of correct and incorrect answers. (Be sure to challenge yourself by choosing a large number!)

To use these programs you first enter the menu program and two common subroutines. Figure 7.1 shows the screen display for the menu program. Figure 7.2 is the program listing of the menu program.



```
1 = ADDITION
2 = SUBTRACTION
3 = MULTIPLICATION
4 = DIVISION




---------------------------------------

CHOOSE PROGRAM: █
```

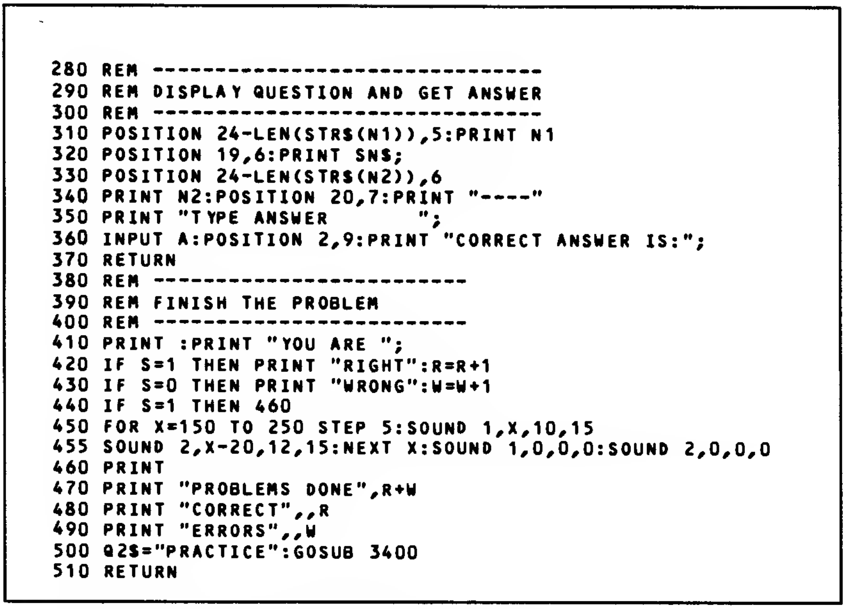
Figure 7.1: Screen Display: Mathematics Practice I Menu

```
100 REM --------------------------
110 REM MATHEMATICS PRACTICE I
120 REM --------------------------
130 GOSUB 6200:DIM SN$(2):REM INITIALIZE
140 K=1
150 X1$="ADDITION"
160 X2$="SUBTRACTION"
170 X3$="MULTIPLICATION"
180 X4$="DIVISION"
190 N=4:GOSUB 7000
200 R=0:W=0
210 Q1$="WHAT IS THE LARGEST NUMBER"
220 Q2$="YOU WOULD LIKE TO"
230 Q3$="PRACTICE WITH"
240 GOSUB 5000:REM ASK QUESTIONS
250 PRINT "   ";:INPUT Y
260 ON X GOSUB 520,660,830,970
270 GOTO 150
```

Figure 7.2: Program Listing: Mathematics Practice I Menu

MATHEMATICS PRACTICE I SUBROUTINES

The Mathematics Practice I programs use two subroutines to display the questions and print and tabulate the answers. You must enter them before you enter the actual drills. We use these subroutines to greatly shorten the programs that follow. The subroutines generate problems with a random number generator, keep track of the total problems done and the total number of correct and incorrect answers, and generate the sound. The subroutines are listed in Figure 7.3.



```
280 REM -------------------------------
290 REM DISPLAY QUESTION AND GET ANSWER
300 REM -------------------------------
310 POSITION 24-LEN(STR$(N1)),5:PRINT N1
320 POSITION 19,6:PRINT SN$;
330 POSITION 24-LEN(STR$(N2)),6
340 PRINT N2:POSITION 20,7:PRINT "----"
350 PRINT "T YPE ANSWER       ";
360 INPUT A:POSITION 2,9:PRINT "CORRECT ANSWER IS:";
370 RETURN
380 REM -------------------------
390 REM FINISH THE PROBLEM
400 REM -------------------------
410 PRINT :PRINT "YOU ARE ";
420 IF S=1 THEN PRINT "RIGHT":R=R+1
430 IF S=0 THEN PRINT "WRONG":W=W+1
440 IF S=1 THEN 460
450 FOR X=150 TO 250 STEP 5:SOUND 1,X,10,15
455 SOUND 2,X-20,12,15:NEXT X:SOUND 1,0,0,0:SOUND 2,0,0,0
460 PRINT
470 PRINT "PROBLEMS DONE",R+W
480 PRINT "CORRECT",,R
490 PRINT "ERRORS",,W
500 Q2$="PRACTICE":GOSUB 3400
510 RETURN
```

Figure 7.3: Program Listing: Mathematics Practice I Subroutines

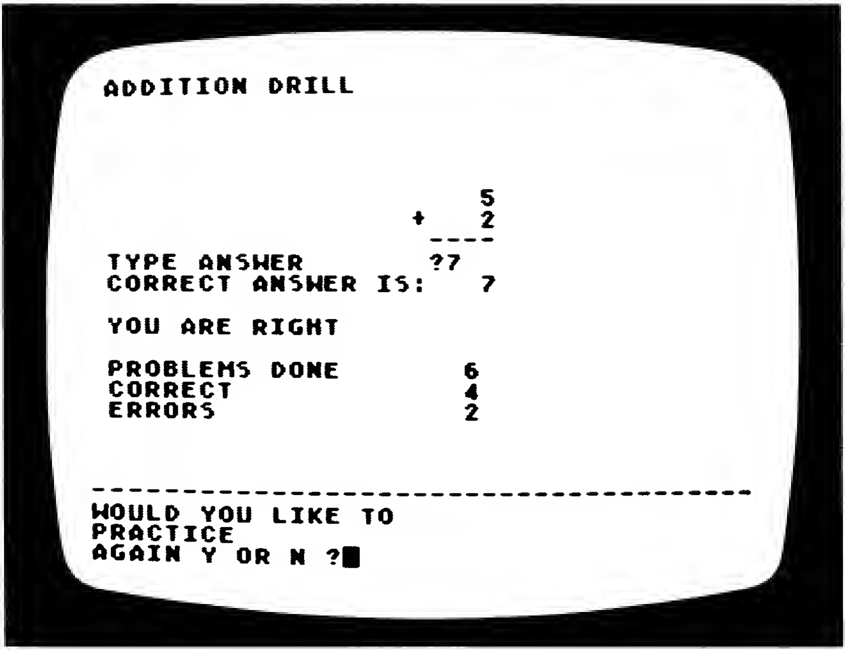
ADDITION

Description

The addition drill presents two random numbers at a time. You must enter the answer from left to right, which will sometimes mean that you must first work out the problem on a piece of paper. The program selects the largest number you work with from the answer you enter when the menu first appears.

Example

The sample display in Figure 7.4 shows an incorrect addition. Note that six problems have been done; four out of six answers have been correct. The Addition program is listed in Figure 7.5.



```
ADDITION DRILL



                            5
                        +   2
                         ----
TYPE ANSWER              ?7
CORRECT ANSWER IS:          7

YOU ARE RIGHT

PROBLEMS DONE             6
CORRECT                   4
ERRORS                    2


----------------------------------------
WOULD YOU LIKE TO
PRACTICE
AGAIN Y OR N ?█
```

Figure 7.4: Screen Display: Addition

```
520 REM -------------------------
530 N$="ADDITION DRILL"
540 REM -------------------------
550 GRAPHICS 0:PRINT N$:PRINT
560 N1=INT(RND(Y)*Y)
570 N2=INT(RND(Y)*Y)
580 SN$="+"
590 GOSUB 280:REM DISP QUEST. GET ANS
600 S=0
610 IF A=N1+N2 THEN S=1
620 POSITION 24-LEN(STR$(N1+N2)),9:PRINT N1+N2
630 GOSUB 380:REM FINISH UP
640 IF YN$="N" THEN RETURN
650 GOTO 520
```

Figure 7.5: Program Listing: Addition

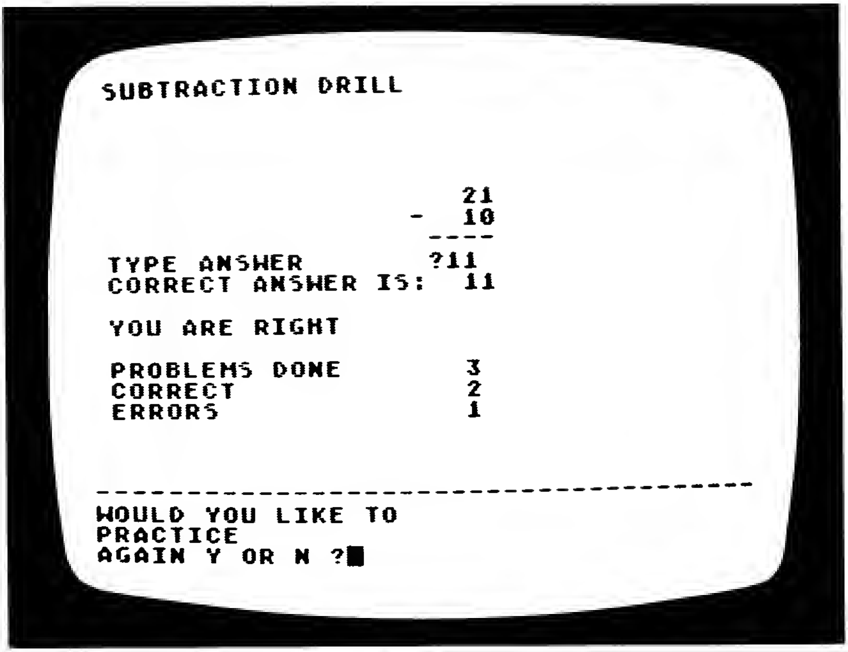
SUBTRACTION

Description

The subtraction drill is similar to the addition drill. The program's selection method for the two numbers guarantees that a positive answer always results. Remember, practice makes perfect—keep trying with new problems.

Example

The screen display in Figure 7.6 shows an incorrect subtraction. Note that two out of three problems have been correctly answered. Figure 7.7 displays the program listing for the subtraction drill.



```
SUBTRACTION DRILL



                            21
                        -   10
                          ----
TYPE ANSWER              ?11
CORRECT ANSWER IS:        11

YOU ARE RIGHT

PROBLEMS DONE             3
CORRECT                   2
ERRORS                    1


------------------------------------
WOULD YOU LIKE TO
PRACTICE
AGAIN Y OR N ?█
```

Figure 7.6: Screen Display: Subtraction

```
660 REM -------------------------
670 N$="SUBTRACTION DRILL"
680 REM -------------------------
690 GRAPHICS 0:PRINT N$:PRINT
700 N1=INT(RND(Y)*Y)
710 N2=INT(RND(Y)*Y)
720 N3=N1
730 IF N1>N2 THEN 750
740 N1=N2:N2=N3
750 SN$="-"
760 GOSUB 280:REM DISP QEST, GET ANS
770 S=0
780 IF A=N1-N2 THEN S=1
790 POSITION 24-LEN(STR$(N1-N2)),9:PRINT N1-N2
800 GOSUB 380:REM FINISH UP
810 IF YN$="N" THEN RETURN
820 GOTO 660
```

Figure 7.7: Program Listing: Subtraction

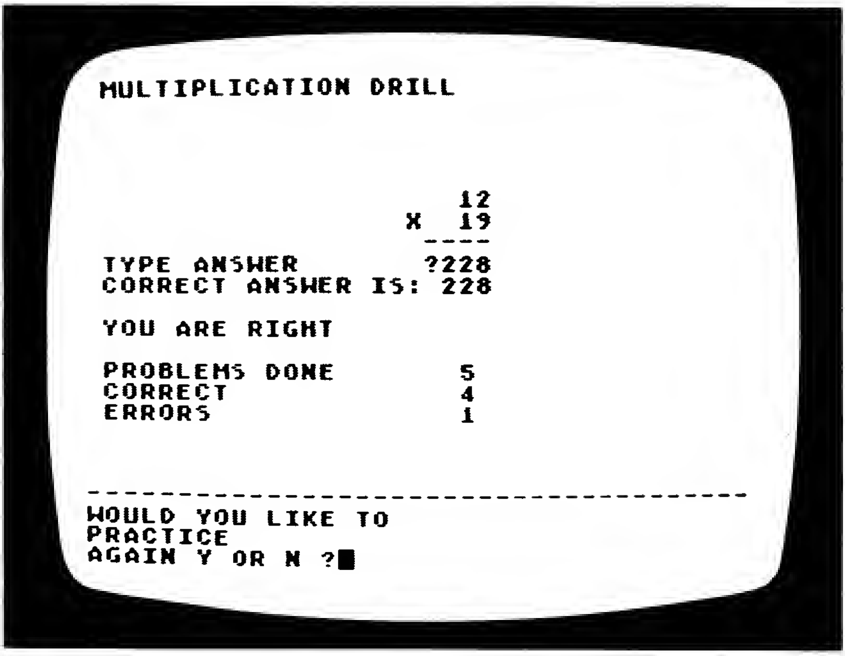
MULTIPLICATION

Description

This program allows you to practice multiplying two numbers. The program generates random numbers; you choose the largest number you want to work with at the time you select the multiplication drill from the menu. This program displays the answer on a single line. If you are multiplying by two or more digits, you may wish to use the Long Multiplication program, which appears later in this chapter. The Long Multiplication program displays the intermediate steps necessary to compute the answer.

Example

Figure 7.8 shows a screen display for the Multiplication program. This display shows that four out of five problems have been done correctly. Figure 7.9 is the program listing for the multiplication drill.



```
MULTIPLICATION DRILL



                         12
                      X  19
                         ----
TYPE ANSWER            ?228
CORRECT ANSWER IS:  228

YOU ARE RIGHT

PROBLEMS DONE          5
CORRECT                4
ERRORS                 1


------------------------------------
WOULD YOU LIKE TO
PRACTICE
AGAIN Y OR N ?█
```

Figure 7.8: Screen Display: Multiplication

```
830 REM -------------------------
840 N$="MULTIPLICATION DRILL"
850 REM -------------------------
860 GRAPHICS 0:PRINT N$:PRINT
870 N1=INT(RND(Y)*Y)
880 N2=INT(RND(Y)*Y)
890 SN$="X"
900 GOSUB 280:REM DISP QUEST, GET ANS
910 S=0
920 IF A=N1*N2 THEN S=1
930 POSITION 24-LEN(STR$(N1*N2)),9:PRINT N1*N2
940 GOSUB 380:REM FINISH UP
950 IF YN$="N" THEN RETURN
960 GOTO 830
```

Figure 7.9: Program Listing: Multiplication

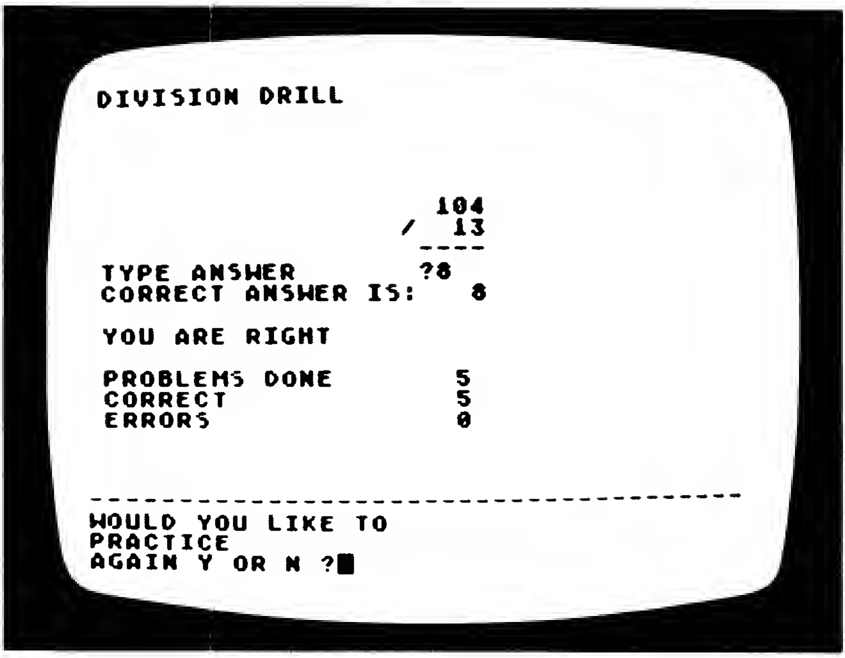
DIVISION

Description

The fourth mathematics drill allows you to practice dividing two whole numbers. Again, you choose the largest number you want to divide by (the denominator) when the Division program is selected from the Mathematics Practice I Menu. The program generates numbers which guarantee that a whole number results from your division. Note that the largest number you work with is the largest answer in this program.

Example

Figure 7.10 shows a screen display for the Division program. Note that this is the fifth problem completed, and that all the answers have been correct. The listing for the Division program appears in Figure 7.11.



```
DIVISION DRILL



                   104
                 /  13
                  ----
 TYPE ANSWER      ?8
 CORRECT ANSWER IS:   8

 YOU ARE RIGHT

 PROBLEMS DONE        5
 CORRECT              5
 ERRORS               0


----------------------------------------
WOULD YOU LIKE TO
PRACTICE
AGAIN Y OR N ?█
```

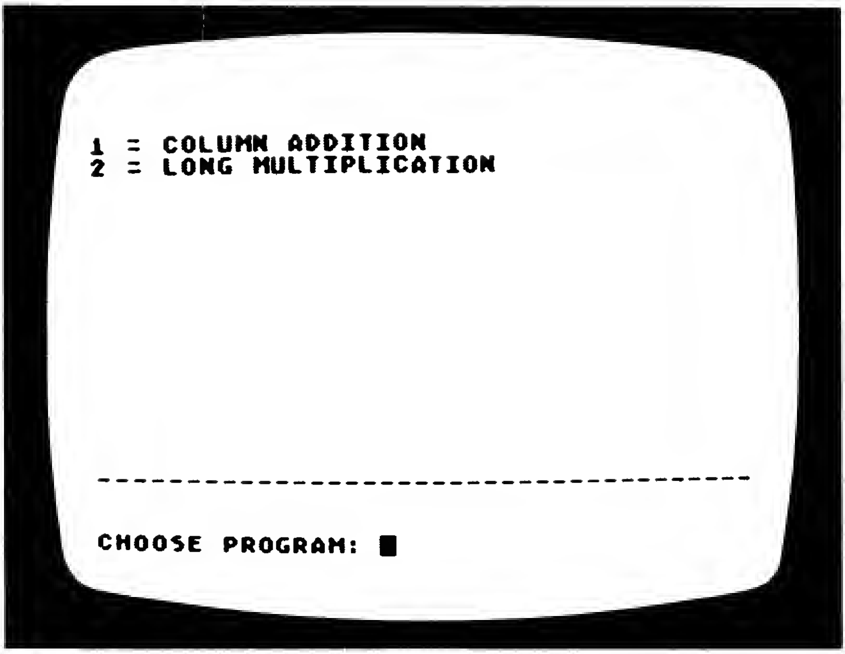
Figure 7.10: Screen Display: Division

```
970 REM -------------------------
980 N$="DIVISION DRILL"
990 REM -------------------------
1000 GRAPHICS 0:PRINT N$:PRINT
1010 N1=INT(RND(Y)*Y)
1020 N2=INT(RND(Y)*Y):IF N2=0 THEN 1020
1030 N1=N1*N2
1040 SN$="/"
1050 GOSUB 280:REM DISP QUEST, GET ANS
1060 S=0
1070 IF A=N1/N2 THEN S=1
1080 POSITION 24-LEN(STR$(N1/N2)),9:PRINT N1/N2
1090 GOSUB 380:REM FINISH UP
1100 IF YN$="N" THEN RETURN
1110 GOTO 970
```

Figure 7.11: Program Listing: Division

MATHEMATICS PRACTICE II MENU

Mathematics Practice II consists of two advanced mathematics drills. The first program allows you to practice adding numbers in columns. The second program allows you to practice multiplying large numbers using long multiplication. For both programs, work out the answer with pencil and paper and then enter it into the computer. Note that the Multiplication program provides all intermediate steps. Figure 7.12 is a screen display of the Mathematics Practice II menu. Figure 7.13 is a listing of the menu program.



```
1 = COLUMN ADDITION
2 = LONG MULTIPLICATION







--------------------------------------

CHOOSE PROGRAM: █
```

Figure 7.12: Screen Display: Mathematics Practice II Menu

```
100 REM -------------------------
110 REM MATHEMATICS PRACTICE II
120 REM -------------------------
130 GOSUB 6200:REM INITIALIZE
140 X1$="COLUMN ADDITION"
150 X2$="LONG MULTIPLICATION"
160 REM SET UP MENU AND DISPLAY IT
170 N=2:GOSUB 7000:REM MENU
180 ON X GOSUB 200,610
190 GOTO 140
```

Figure 7.13: Program Listing: Mathematics Practice II Menu

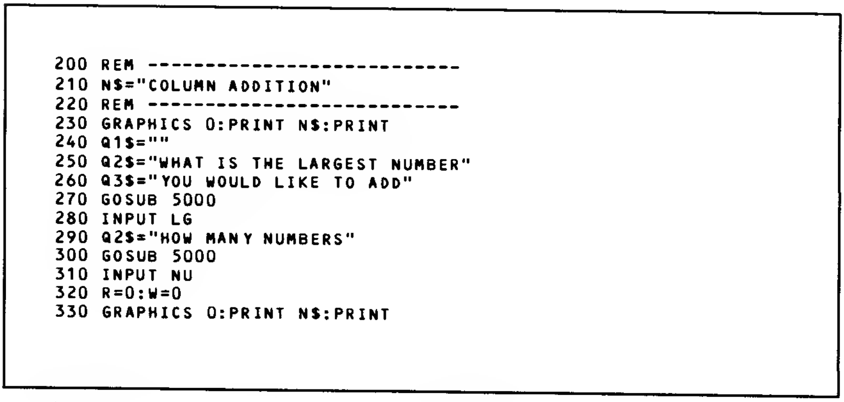
COLUMN ADDITION

Description

The Column Addition program allows you to practice adding columns of numbers. You select this drill by depressing the number 1 from the menu program. The program then asks you to enter both the largest number and how many numbers you would like to add. The program keeps track of correct and incorrect answers.

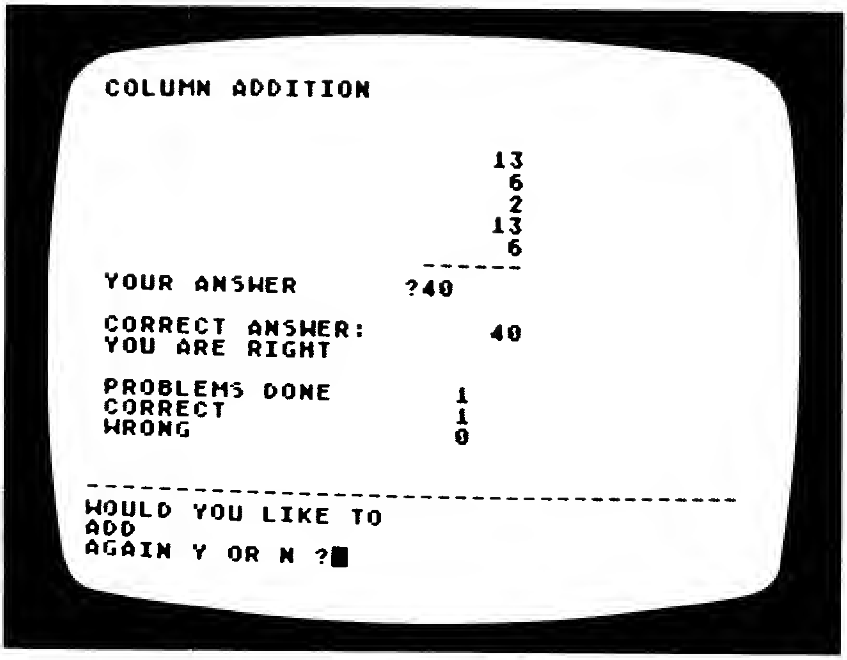
Example

Figure 7.14 shows the correct addition of five numbers. Figure 7.15 is the program listing for this mathematics drill.



```
200 REM --------------------------
210 N$="COLUMN ADDITION"
220 REM --------------------------
230 GRAPHICS 0:PRINT N$:PRINT
240 Q1$=""
250 Q2$="WHAT IS THE LARGEST NUMBER"
260 Q3$="YOU WOULD LIKE TO ADD"
270 GOSUB 5000
280 INPUT LG
290 Q2$="HOW MANY NUMBERS"
300 GOSUB 5000
310 INPUT NU
320 R=0:W=0
330 GRAPHICS 0:PRINT N$:PRINT
```

Figure 7.15: Program Listing: Column Addition (continues)



```
COLUMN ADDITION


                                        13
                                         6
                                         2
                                        13
                                         6
                                   ------
YOUR ANSWER                  ?40

CORRECT ANSWER:                         40
YOU ARE RIGHT

PROBLEMS DONE                    1
CORRECT                          1
WRONG                            0

----------------------------------------
WOULD YOU LIKE TO
ADD
AGAIN Y OR N ?
```

Figure 7.14: Screen Display: Column Addition

```
340 TOT=0
350 FOR I=1 TO NU
360 N1=INT(RND(LG)*LG)
370 TOT=TOT+N1
380 POSITION 26-LEN(STR$(N1)),I+2:PRINT N1
390 NEXT I
400 POSITION 20,I+2:PRINT "------"
420 PRINT "YOUR ANSWER      ";
430 INPUT A
44U POSITION 2,I+5:PRINT "CORRECT ANSWER:";
450 POSITION 26-LEN(STR$(TOT)),I+5:PRINT TOT
470 PRINT "YOU ARE ";
480 IF TOT=A THEN PRINT "RIGHT":R=R+1:GOTO 540
490 PRINT "WRONG":W=W+1
500 REM ADD SOME SOUND
510 FOR X=150 TO 255 STEP 5:SOUND 0,X,10,15:SOUND 1,X-50,8,9
520 SOUND 2,INT(X/40)*40-50,10,12:NEXT X
530 SOUND 1,0,0,0:SOUND 2,0,0,0:SOUND 0,0,0,0
540 PRINT
550 PRINT "PROBLEMS DONE",R+W
560 PRINT "CORRECT",,R
570 PRINT "WRONG",,W
580 Q2$="ADD":GOSUB 3400
590 IF YN$="N" THEN RETURN
600 GOTO 330
```

Figure 7.15: Program Listing: Column Addition

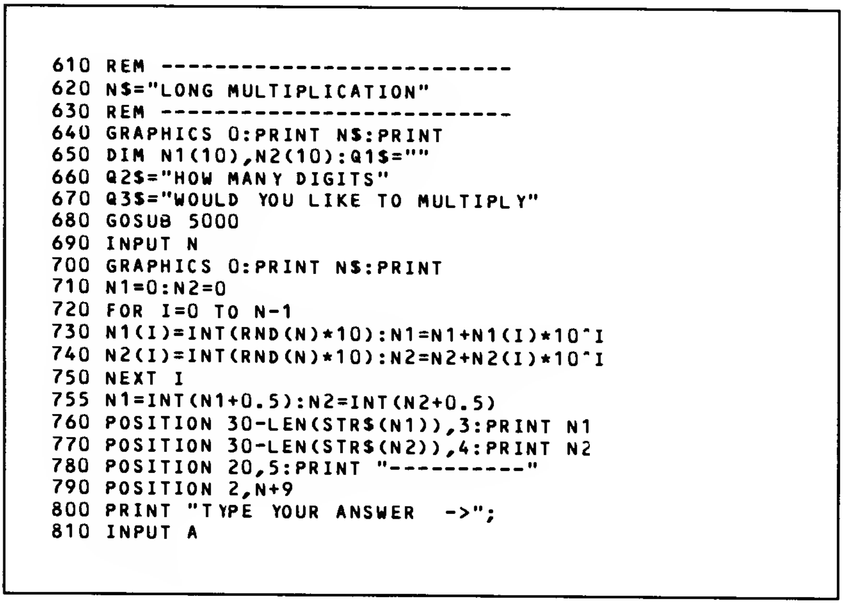
LONG MULTIPLICATION

Description

We now present a program that allows you to practice multiplying numbers using long multiplication. You select this program from the Mathematics Practice II Menu by pressing the 2 on your keyboard. You then enter the number of digits you wish to work with. The program presents you with two numbers to multiply; you work out your answer by hand and enter the final answer. The screen displays the correct answer and the steps that are necessary to compute the answer. By reviewing the various steps, you can see exactly where you make your mistakes.

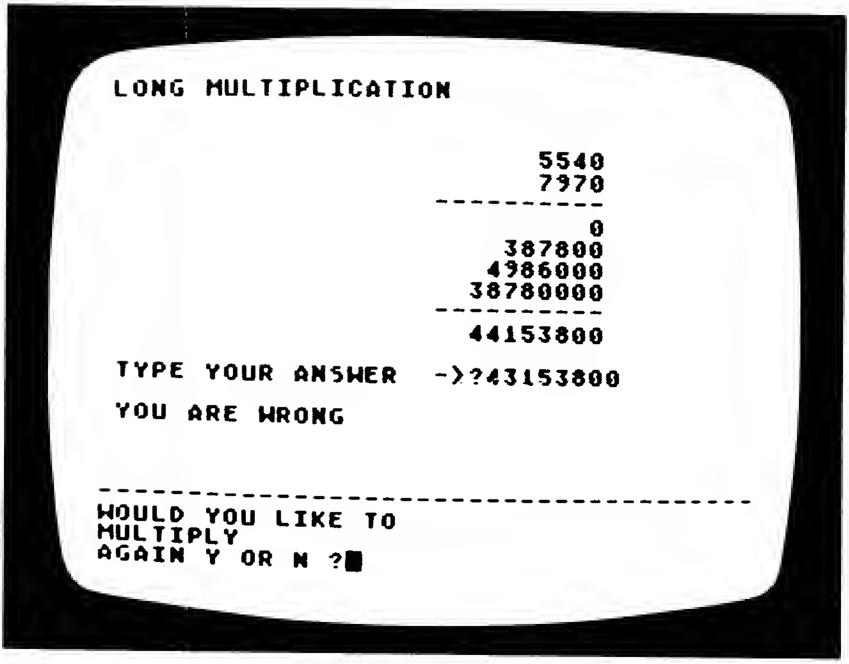
Example

Figure 7.16 shows the result of multiplying two large numbers. Note that all of the intermediate results are displayed. Figure 7.17 is the Long Multiplication program listing.



```
610 REM -------------------------
620 N$="LONG MULTIPLICATION"
630 REM -------------------------
640 GRAPHICS 0:PRINT N$:PRINT
650 DIM N1(10),N2(10):Q1$=""
660 Q2$="HOW MANY DIGITS"
670 Q3$="WOULD YOU LIKE TO MULTIPLY"
680 GOSUB 5000
690 INPUT N
700 GRAPHICS 0:PRINT N$:PRINT
710 N1=0:N2=0
720 FOR I=0 TO N-1
730 N1(I)=INT(RND(N)*10):N1=N1+N1(I)*10^I
740 N2(I)=INT(RND(N)*10):N2=N2+N2(I)*10^I
750 NEXT I
755 N1=INT(N1+0.5):N2=INT(N2+0.5)
760 POSITION 30-LEN(STR$(N1)),3:PRINT N1
770 POSITION 30-LEN(STR$(N2)),4:PRINT N2
780 POSITION 20,5:PRINT "----------"
790 POSITION 2,N+9
800 PRINT "TYPE YOUR ANSWER  ->";
810 INPUT A
```

Figure 7.17: Program Listing: Long Multiplication (continues)



```
LONG MULTIPLICATION

                                  5540
                                  7970
                            ----------
                                     0
                                387800
                               4986000
                              38780000
                            ----------
                              44153800
TYPE YOUR ANSWER    ->?43153800

YOU ARE WRONG


-----------------------------------
WOULD YOU LIKE TO
MULTIPLY
AGAIN Y OR N ?
```

Figure 7.16: Screen Display: Long Multiplication

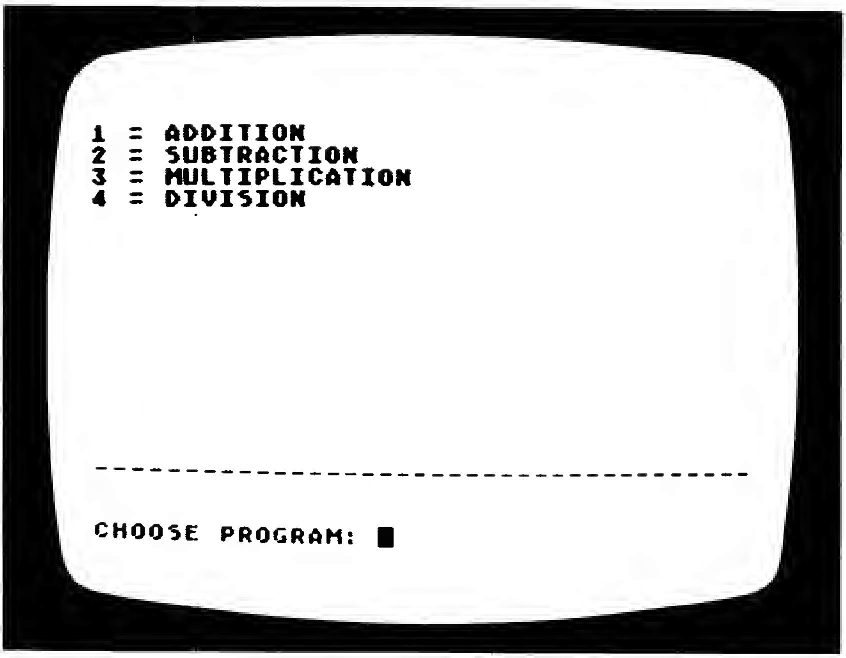
```
820 FOR I=0 TO N-1
830 X=N1*N2(I)*10^I
835 X=INT(X+0.5)
840 POSITION 30-LEN(STR$(X)),I+6:PRINT X
850 NEXT I
860 POSITION 20,N+6:PRINT "----------"
870 POSITION 30-LEN(STR$(N1*N2)),N+7
880 PRINT N1*N2
890 PRINT :PRINT :PRINT
900 PRINT "YOU ARE ";
910 IF A=N1*N2 THEN 970
920 PRINT "WRONG":FOR X=200 TO 255
930 SOUND 1,X,10,10:SOUND 2,X-30,10,10
940 SOUND 3,X-INT(X/10)*10+40,10,10
950 NEXT X:SOUND 1,0,0,0:SOUND 2,0,0,0
960 SOUND 3,0,0,0:GOTO 1000
970 PRINT "RIGHT":FOR X=90 TO 50 STEP -4
980 SOUND 1,X,10,6:NEXT X
990 SOUND 1,0,0,0
1000 Q2$="MULTIPLY":GOSUB 3400
1010 IF YN$="N" THEN RETURN
1020 GOTO 700
```

Figure 7.17: Program Listing: Long Multiplication

MATHEMATICS PRACTICE III MENU

The final set of mathematics drills allows you to practice adding, subtracting, multiplying, and dividing fractions. You work out your answers by hand and enter the final answer. All answers in this section are reduced to least-common-denominator form; conversions to mixed numbers are not made. Each program keeps track of your correct and incorrect answers.

The four fraction programs work from a common menu. Figure 7.18 displays the Mathematics Practice III Menu; Figure 7.19 is the listing of the menu program.



```
1 = ADDITION
2 = SUBTRACTION
3 = MULTIPLICATION
4 = DIVISION




----------------------------------------

CHOOSE PROGRAM: █
```

Figure 7.18: Screen Display: Mathematics Practice III Menu

```
100 REM --------------------------
110 REM MATHEMATICS PRACTICE III
120 REM --------------------------
130 GOSUB 6200:DIM SN$(2):REM INITIALIZE
140 X1$="ADDITION"
150 X2$="SUBTRACTION"
160 X3$="MULTIPLICATION"
170 X4$="DIVISION"
180 N=4:GOSUB 7000
190 R=0:W=0
200 Q1$="WHAT IS THE LARGEST DENOMINATOR"
210 Q2$="YOU WOULD LIKE TO"
220 Q3$="PRACTICE WITH   "
230 GOSUB 5000
240 INPUT Y
250 ON X GOSUB 860,1010,1170,1320
260 GOTO 140
```

Figure 7.19: Program Listing: Mathematics Practice III Menu

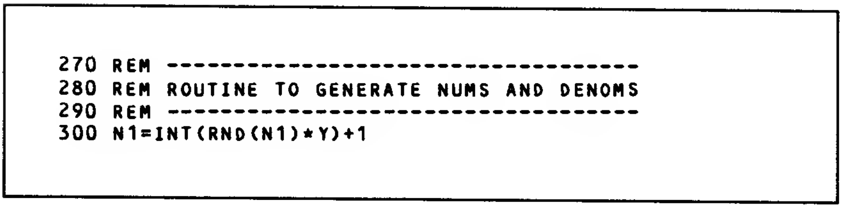
FRACTION SUBROUTINES

Prior to entering the fraction programs, you should enter the four subroutines shown in Figure 7.20. These subroutines generate the fractions and the greatest common denominator, display the question, and print the answer.

The subroutines use a random number generator to select the numerators and denominators. They also display the problem on the screen in fraction form, as shown below:

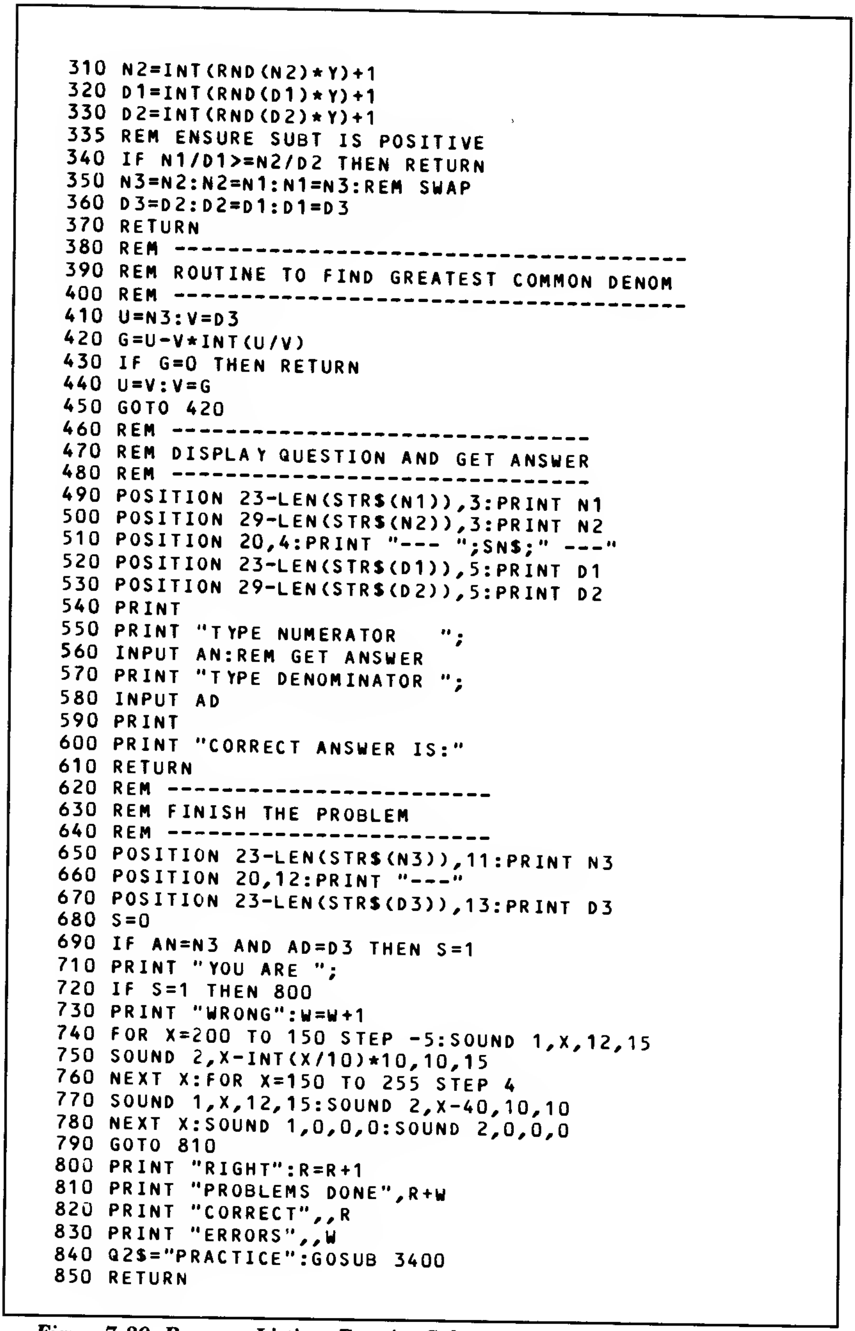
$$\frac{4}{5} + \frac{5}{6}$$

When you answer, the subroutines ask you for the numerator and denominator separately.



```
270 REM -----------------------------------
280 REM ROUTINE TO GENERATE NUMS AND DENOMS
290 REM -----------------------------------
300 N1=INT(RND(N1)*Y)+1
```

Figure 7.20: Program Listing: Fraction Subroutines (continues)



```
310 N2=INT(RND(N2)*Y)+1
320 D1=INT(RND(D1)*Y)+1
330 D2=INT(RND(D2)*Y)+1
335 REM ENSURE SUBT IS POSITIVE
340 IF N1/D1>=N2/D2 THEN RETURN
350 N3=N2:N2=N1:N1=N3:REM SWAP
360 D3=D2:D2=D1:D1=D3
370 RETURN
380 REM ------------------------------------
390 REM ROUTINE TO FIND GREATEST COMMON DENOM
400 REM ------------------------------------
410 U=N3:V=D3
420 G=U-V*INT(U/V)
430 IF G=0 THEN RETURN
440 U=V:V=G
450 GOTO 420
460 REM ------------------------------
470 REM DISPLAY QUESTION AND GET ANSWER
480 REM ------------------------------
490 POSITION 23-LEN(STR$(N1)),3:PRINT N1
500 POSITION 29-LEN(STR$(N2)),3:PRINT N2
510 POSITION 20,4:PRINT "--- ";SN$;" ---"
520 POSITION 23-LEN(STR$(D1)),5:PRINT D1
530 POSITION 29-LEN(STR$(D2)),5:PRINT D2
540 PRINT
550 PRINT "TYPE NUMERATOR   ";
560 INPUT AN:REM GET ANSWER
570 PRINT "TYPE DENOMINATOR ";
580 INPUT AD
590 PRINT
600 PRINT "CORRECT ANSWER IS:"
610 RETURN
620 REM ----------------------
630 REM FINISH THE PROBLEM
640 REM ----------------------
650 POSITION 23-LEN(STR$(N3)),11:PRINT N3
660 POSITION 20,12:PRINT "---"
670 POSITION 23-LEN(STR$(D3)),13:PRINT D3
680 S=0
690 IF AN=N3 AND AD=D3 THEN S=1
710 PRINT "YOU ARE ";
720 IF S=1 THEN 800
730 PRINT "WRONG":W=W+1
740 FOR X=200 TO 150 STEP -5:SOUND 1,X,12,15
750 SOUND 2,X-INT(X/10)*10,10,15
760 NEXT X:FOR X=150 TO 255 STEP 4
770 SOUND 1,X,12,15:SOUND 2,X-40,10,10
780 NEXT X:SOUND 1,0,0,0:SOUND 2,0,0,0
790 GOTO 810
800 PRINT "RIGHT":R=R+1
810 PRINT "PROBLEMS DONE",R+W
820 PRINT "CORRECT",,R
830 PRINT "ERRORS",,W
840 Q2$="PRACTICE":GOSUB 3400
850 RETURN
```

Figure 7.20: Program Listing: Fraction Subroutines

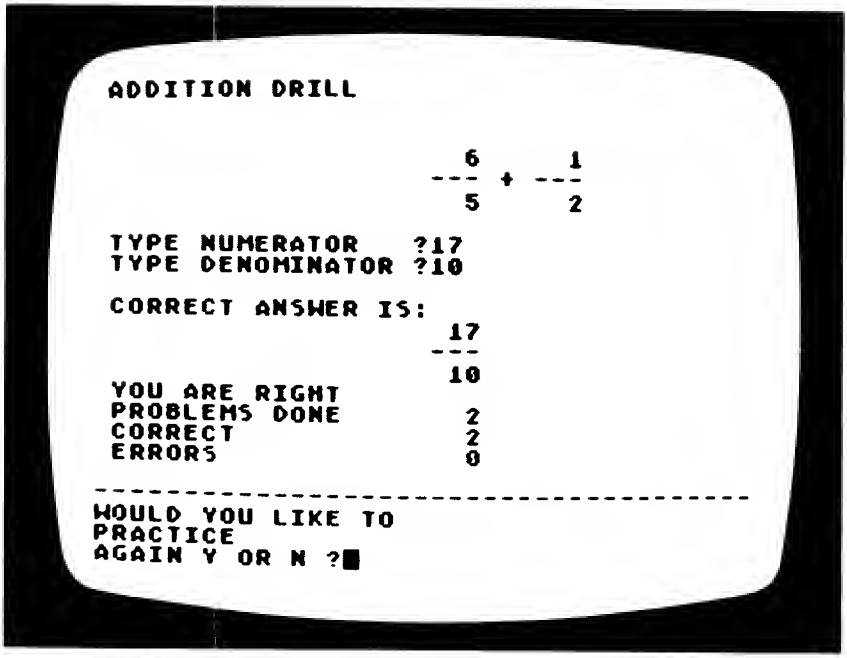
FRACTION ADDITION

Description

The Fraction Addition drill allows you to practice adding two fractions. You select this program from the fraction program menu by striking the 1 on your keyboard. You must also enter the largest denominator you want to work with. (Don't be afraid of large denominators!)

Example

Figure 7.21 shows the addition of two fractions. Note that this is the second problem of a series. Figure 7.22 is the program listing for the addition drill.



```
ADDITION DRILL


                         6     1
                        --- + ---
                         5     2
TYPE NUMERATOR   ?17
TYPE DENOMINATOR ?10

CORRECT ANSWER IS:
                        17
                        ---
                        10
YOU ARE RIGHT
PROBLEMS DONE            2
CORRECT                  2
ERRORS                   0
--------------------------------------
WOULD YOU LIKE TO
PRACTICE
AGAIN Y OR N ?█
```

Figure 7.21: Screen Display: Fraction Addition

```
860 REM ------------------------
870 N$="ADDITION DRILL"
880 REM ------------------------
890 GRAPHICS 0:PRINT N$:PRINT
900 GOSUB 270:REM GENERATE NUMS AND DENOMS
910 SN$="+"
920 GOSUB 460:REM DISP QUEST, GET ANS
930 N3=D2*N1+D1*N2:REM BEFORE CLEANUP
940 D3=D1*D2
950 REM FIND GREATEST COMMON DENOM
960 GOSUB 380
970 N3=N3/V:D3=D3/V
980 GOSUB 620:REM FINISH UP
990 IF YN$="N" THEN RETURN
1000 GOTO 860
```

Figure 7.22: Program Listing: Fraction Addition

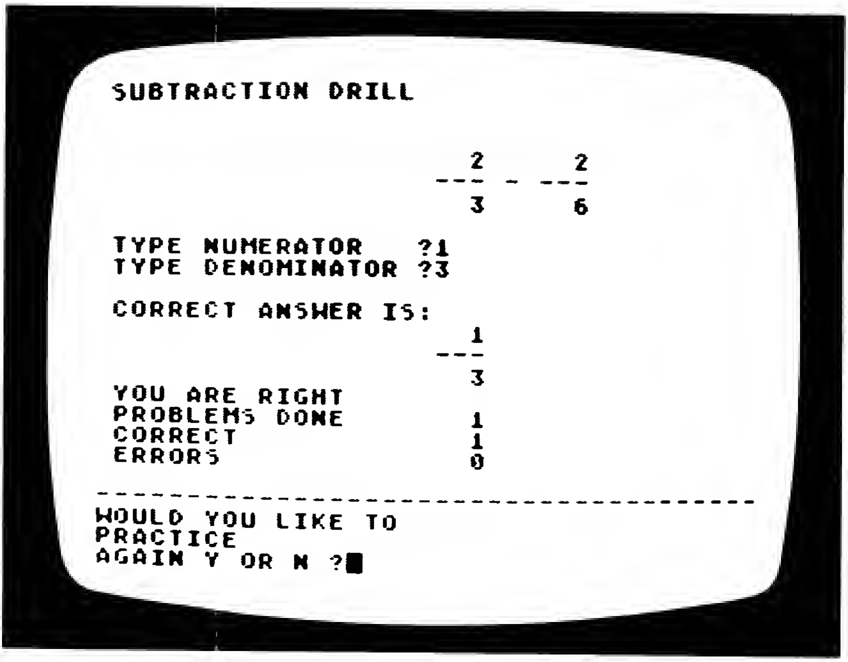
FRACTION SUBTRACTION

Description

You can practice subtracting fractions with this program. You select the program by pressing 2 as you run the fraction drill menu. You also select the largest denominator you want to work with. Like the subtraction drill in the first menu, this program generates its numbers so that a positive answer always results.

Example

Figure 7.23 shows a screen display for the Fraction Subtraction program. Figure 7.24 is the program listing.



```
 SUBTRACTION DRILL


                     2     2
                   --- - ---
                     3     6

 TYPE NUMERATOR   ?1
 TYPE DENOMINATOR ?3

 CORRECT ANSWER IS:
                     1
                   ---
                     3
 YOU ARE RIGHT
 PROBLEMS DONE       1
 CORRECT             1
 ERRORS              0

---------------------------------------
WOULD YOU LIKE TO
PRACTICE
AGAIN Y OR N ?█
```

Figure 7.23: Screen Display: Fraction Subtraction

```
1010 REM ------------------------
1020 N$="SUBTRACTION DRILL"
1030 REM ------------------------
1040 GRAPHICS 0:PRINT N$:PRINT
1050 GOSUB 270:REM GENERATE NUMS AND DENOMS
1060 N3=D2*N1-D1*N2:REM BEFORE CLEANUP
1070 IF N3=0 THEN 1050:REM DISALLOW ZERO
1080 SN$="-"
1090 GOSUB 460:REM DISP QUEST, GET ANS
1100 D3=D1*D2
1110 REM FIND GREATEST COMMON DENOM
1120 GOSUB 380
1130 N3=N3/V:D3=D3/V
1140 GOSUB 620:REM FINISH UP
1150 IF YN$="N" THEN RETURN
1160 GOTO 1010
```

Figure 7.24: Program Listing: Fraction Subtraction

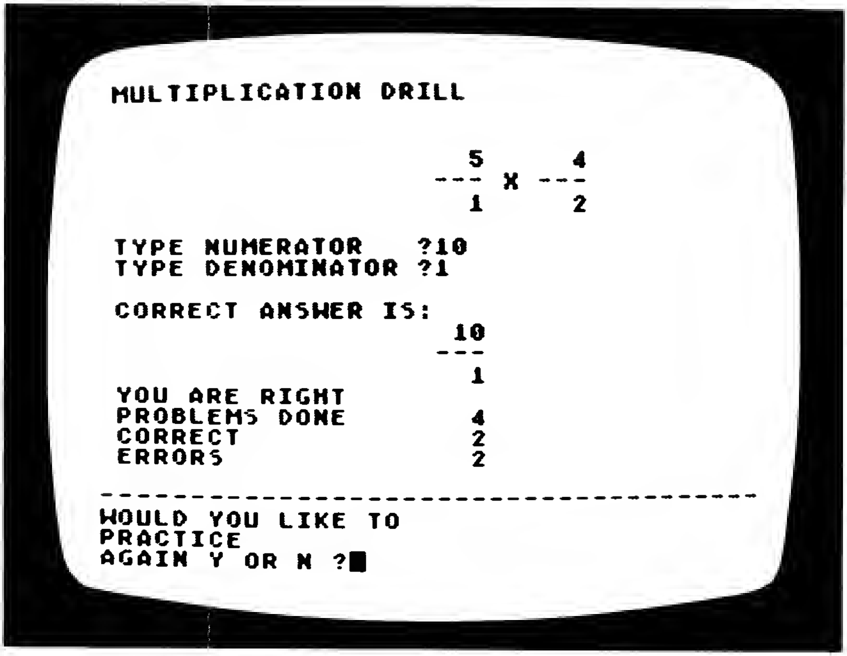
FRACTION MULTIPLICATION

Description

The Fraction Multiplication program allows you to practice multiplying two fractions. You select this program from the fraction menu by pressing the number 3. Fraction multiplication is very easy to learn, and it comes in handy—don't be afraid to try this drill.

Example

Figure 7.25 shows a screen display for the Fraction Multiplication program. Note that two out of four problems have been correctly answered. Figure 7.26 is the Fraction Multiplication program listing.



```
MULTIPLICATION DRILL


                     5     4
                    --- X ---
                     1     2

TYPE NUMERATOR   ?10
TYPE DENOMINATOR ?1

CORRECT ANSWER IS:
                    10
                    ---
                     1
YOU ARE RIGHT
PROBLEMS DONE        4
CORRECT              2
ERRORS               2

---------------------------------------
WOULD YOU LIKE TO
PRACTICE
AGAIN Y OR N ?█
```

Figure 7.25: Screen Display: Fraction Multiplication

```
1170 REM ------------------------
1180 N$="MULTIPLICATION DRILL"
1190 REM ------------------------
1200 GRAPHICS 0:PRINT N$:PRINT
1210 GOSUB 270:REM GENERATE NUMS AND DENOMS
1220 N3=N1*N2
1230 D3=D1*D2:REM BEFORE CLEANUP
1240 REM FIND GREATEST COMMON DENOM
1250 GOSUB 380
1260 N3=N3/V:D3=D3/V
1270 SN$="X"
1280 GOSUB 460:REM DISP QUEST, GET ANS
1290 GOSUB 620:REM FINISH UP
1300 IF YN$="N" THEN RETURN
1310 GOTO 1170
```

Figure 7.26: Program Listing: Fraction Multiplication

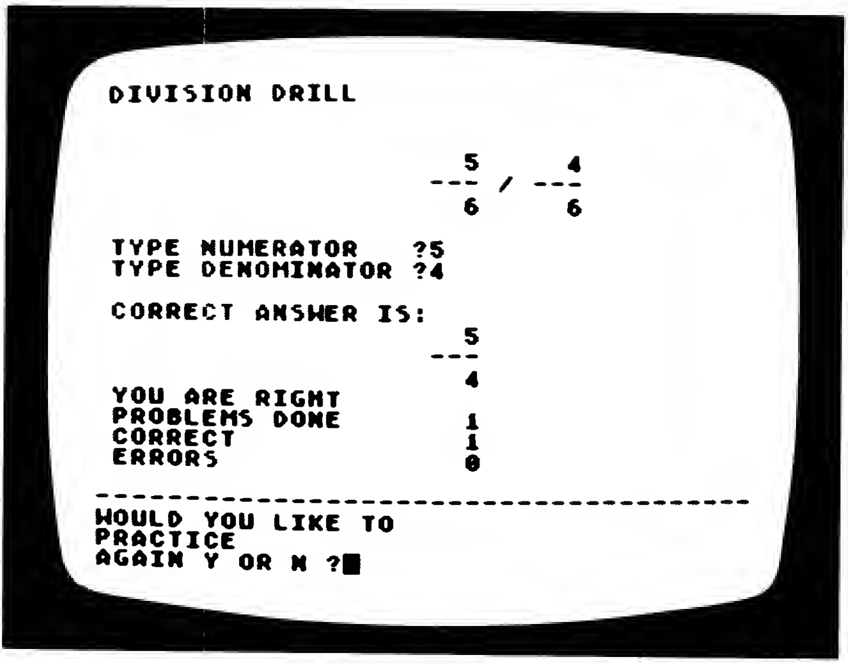
FRACTION DIVISION

Description

The final fraction program allows you to practice dividing fractions. This program is chosen by pressing the 4 on your keyboard from the menu program. Dividing fractions is the same as multiplying fractions, except that you invert the second fraction (the denominator). Practice with this program will sharpen your skills and make fraction division a less formidable task for you.

Example

Figure 7.27 shows a screen display for the Fraction Division program. Figure 7.28 is the program listing.



```
DIVISION DRILL


                       5     4
                      --- / ---
                       6     6

TYPE NUMERATOR   ?5
TYPE DENOMINATOR ?4

CORRECT ANSWER IS:
                       5
                      ---
                       4
YOU ARE RIGHT
PROBLEMS DONE          1
CORRECT                1
ERRORS                 0

--------------------------------------
WOULD YOU LIKE TO
PRACTICE
AGAIN Y OR N ?
```

Figure 7.27: Screen Display: Fraction Division

```
1320 REM -------------------------
1330 N$="DIVISION DRILL"
1340 REM -------------------------
1350 GRAPHICS 0:PRINT N$:PRINT
1360 GOSUB 270:REM GENERATE NUMS AND DENOMS
1370 N3=N1*D2
1380 D3=D1*N2:REM BEFORE CLEANUP
1390 REM FIND GREATEST COMMON DENOM
1400 GOSUB 380
1410 N3=N3/V:D3=D3/V
1420 SN$="/"
1430 GOSUB 460:REM DISPLAY QUEST, GET ANSWER
1440 GOSUB 620:REM FINISH UP
1450 IF YN$="N" THEN RETURN
1460 GOTO 1320
```

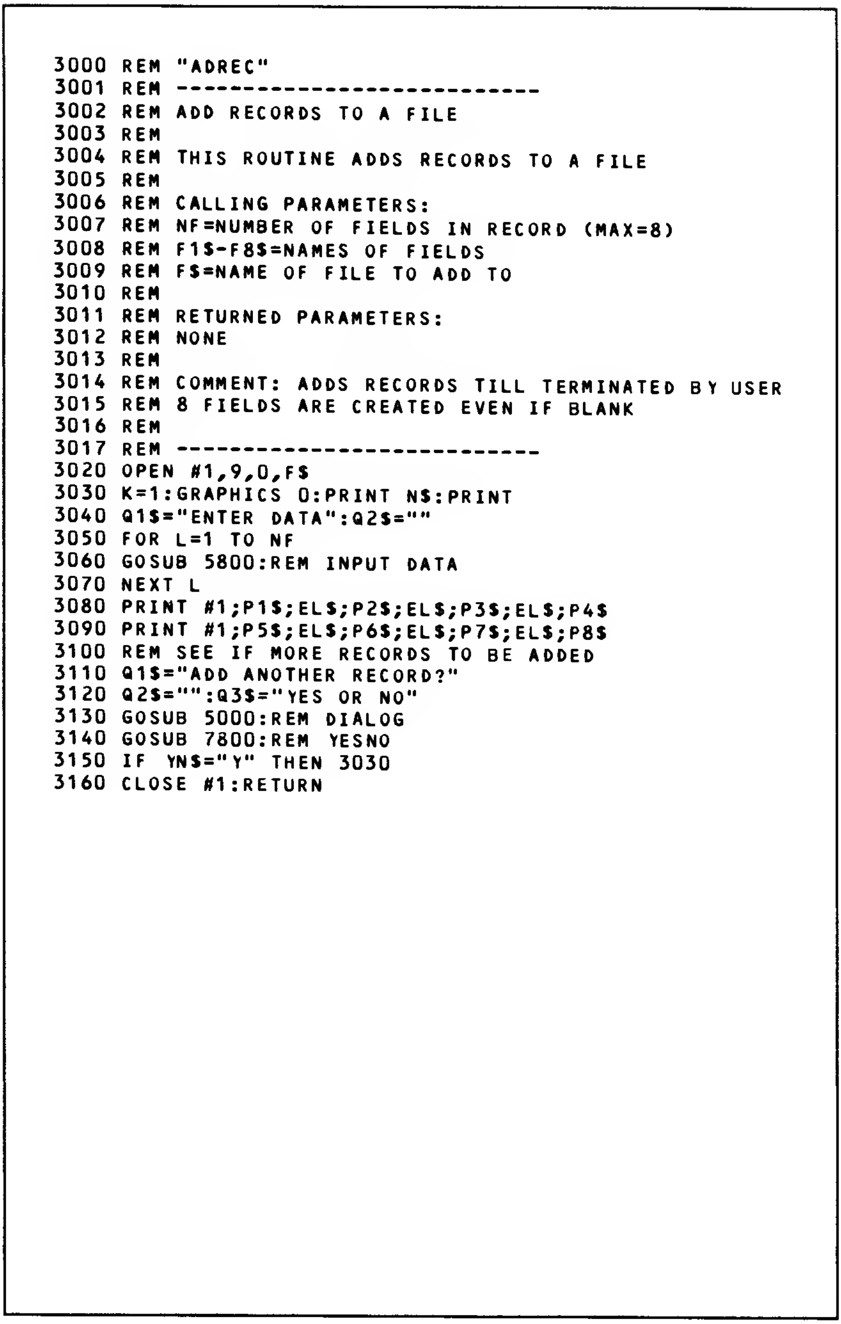
Figure 7.28: Program Listing: Fraction Division

A
CENTRAL SUBROUTINES

These subroutines allow us to create useful programs that are very short and thus easy to enter. Interested users should read Appendix B to see how these subroutines can be used in other applications.

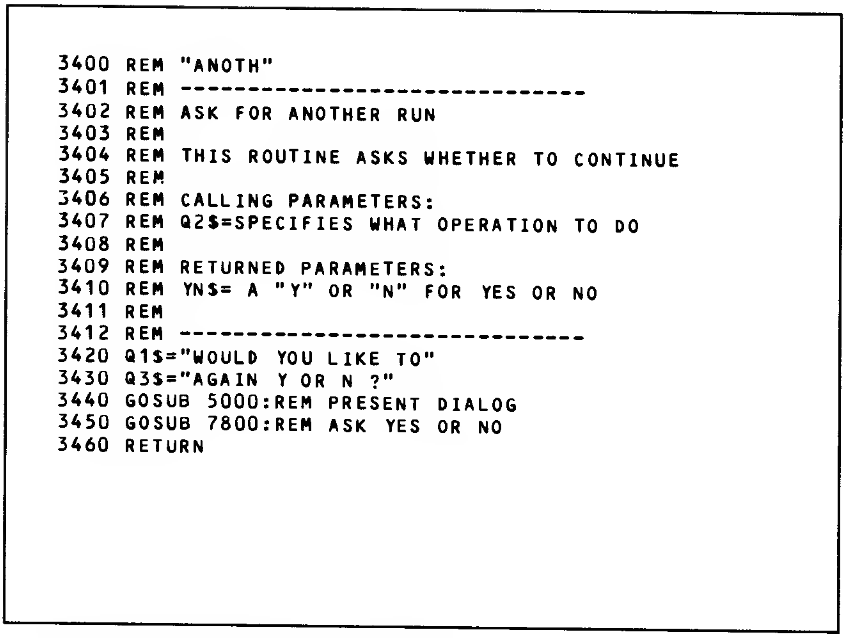
This appendix contains the program listings for the central subroutines used throughout the book. The programs contain many remark statements to help you understand the functions of each subroutine.

You can save retyping these subroutines many times by saving them together in one disk file. Add them to your application programs as explained in Chapter 1. Remember that you need type only the first REM in each subroutine.



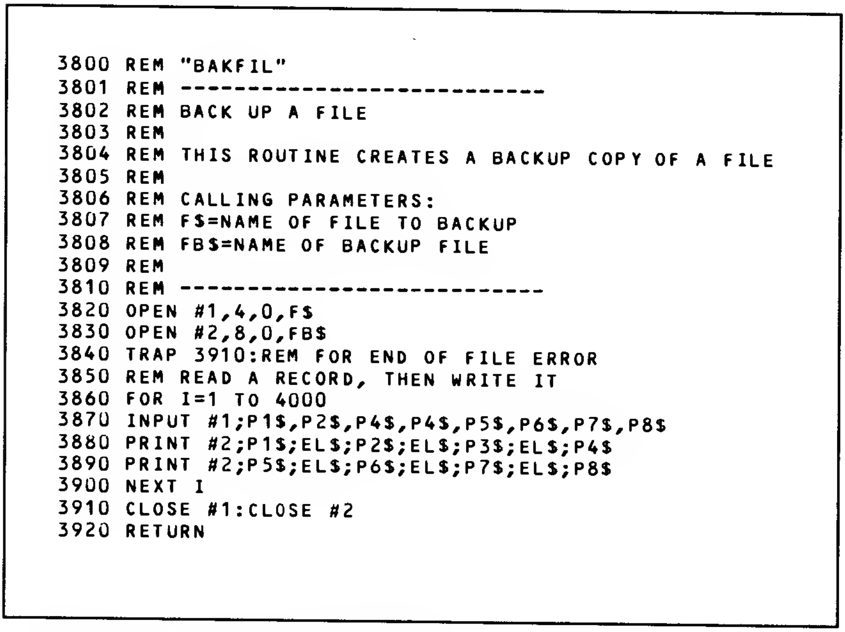
```
3000 REM "ADREC"
3001 REM ---------------------------
3002 REM ADD RECORDS TO A FILE
3003 REM
3004 REM THIS ROUTINE ADDS RECORDS TO A FILE
3005 REM
3006 REM CALLING PARAMETERS:
3007 REM NF=NUMBER OF FIELDS IN RECORD (MAX=8)
3008 REM F1$-F8$=NAMES OF FIELDS
3009 REM F$=NAME OF FILE TO ADD TO
3010 REM
3011 REM RETURNED PARAMETERS:
3012 REM NONE
3013 REM
3014 REM COMMENT: ADDS RECORDS TILL TERMINATED BY USER
3015 REM 8 FIELDS ARE CREATED EVEN IF BLANK
3016 REM
3017 REM ---------------------------
3020 OPEN #1,9,0,F$
3030 K=1:GRAPHICS 0:PRINT N$:PRINT
3040 Q1$="ENTER DATA":Q2$=""
3050 FOR L=1 TO NF
3060 GOSUB 5800:REM INPUT DATA
3070 NEXT L
3080 PRINT #1;P1$;EL$;P2$;EL$;P3$;EL$;P4$
3090 PRINT #1;P5$;EL$;P6$;EL$;P7$;EL$;P8$
3100 REM SEE IF MORE RECORDS TO BE ADDED
3110 Q1$="ADD ANOTHER RECORD?"
3120 Q2$="":Q3$="YES OR NO"
3130 GOSUB 5000:REM DIALOG
3140 GOSUB 7800:REM YESNO
3150 IF YN$="Y" THEN 3030
3160 CLOSE #1:RETURN
```

Figure A.1: Add Record Subroutine



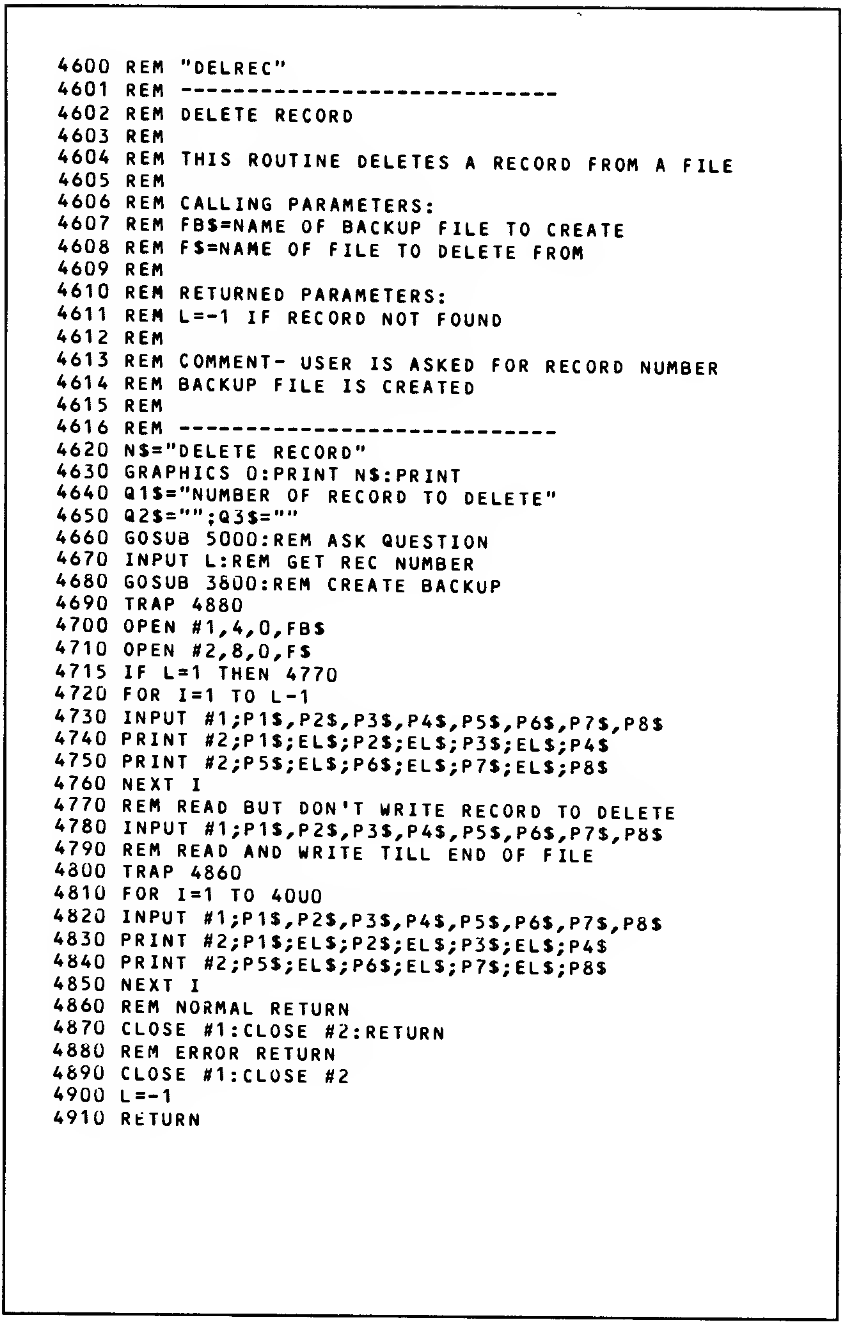
```
3400 REM "ANOTH"
3401 REM ------------------------------
3402 REM ASK FOR ANOTHER RUN
3403 REM
3404 REM THIS ROUTINE ASKS WHETHER TO CONTINUE
3405 REM
3406 REM CALLING PARAMETERS:
3407 REM Q2$=SPECIFIES WHAT OPERATION TO DO
3408 REM
3409 REM RETURNED PARAMETERS:
3410 REM YN$= A "Y" OR "N" FOR YES OR NO
3411 REM
3412 REM ------------------------------
3420 Q1$="WOULD YOU LIKE TO"
3430 Q3$="AGAIN Y OR N ?"
3440 GOSUB 5000:REM PRESENT DIALOG
3450 GOSUB 7800:REM ASK YES OR NO
3460 RETURN
```

Figure A.2: Request to Run Again Subroutine



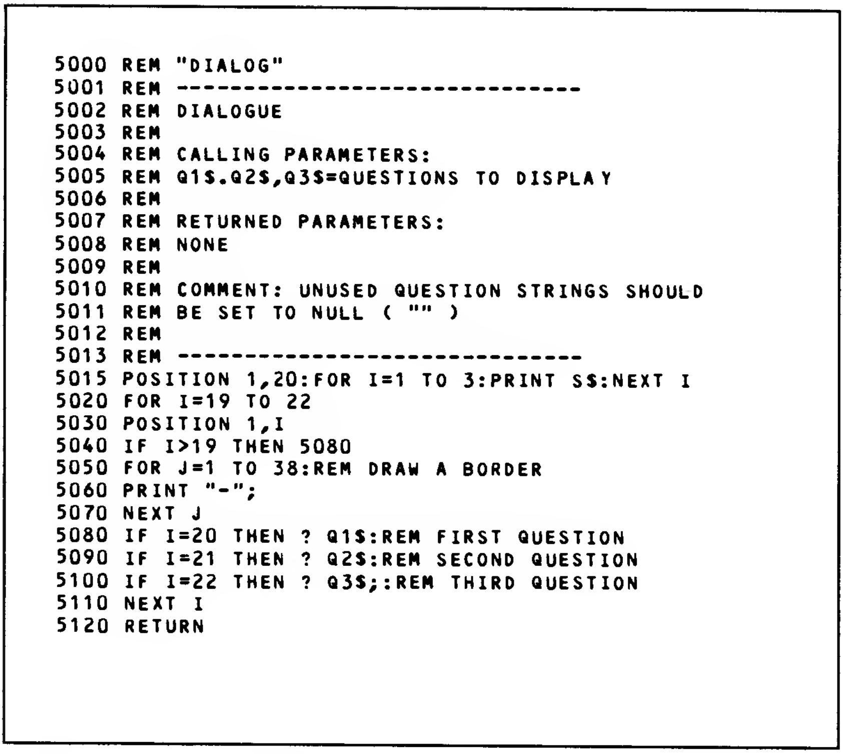
```
3800 REM "BAKFIL"
3801 REM ---------------------------
3802 REM BACK UP A FILE
3803 REM
3804 REM THIS ROUTINE CREATES A BACKUP COPY OF A FILE
3805 REM
3806 REM CALLING PARAMETERS:
3807 REM F$=NAME OF FILE TO BACKUP
3808 REM FB$=NAME OF BACKUP FILE
3809 REM
3810 REM ---------------------------
3820 OPEN #1,4,0,F$
3830 OPEN #2,8,0,FB$
3840 TRAP 3910:REM FOR END OF FILE ERROR
3850 REM READ A RECORD, THEN WRITE IT
3860 FOR I=1 TO 4000
3870 INPUT #1;P1$,P2$,P4$,P4$,P5$,P6$,P7$,P8$
3880 PRINT #2;P1$;EL$;P2$;EL$;P3$;EL$;P4$
3890 PRINT #2;P5$;EL$;P6$;EL$;P7$;EL$;P8$
3900 NEXT I
3910 CLOSE #1:CLOSE #2
3920 RETURN
```

Figure A.3: Create Backup File Subroutine



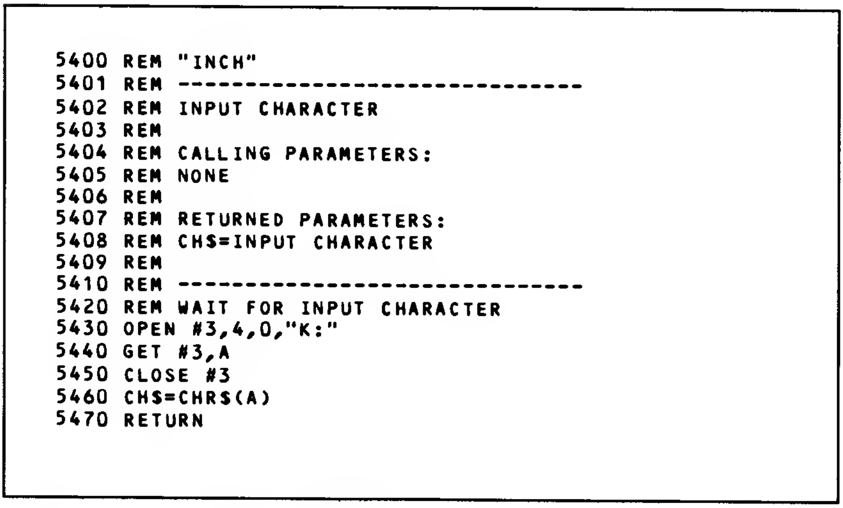
```
4600 REM "DELREC"
4601 REM ----------------------------
4602 REM DELETE RECORD
4603 REM
4604 REM THIS ROUTINE DELETES A RECORD FROM A FILE
4605 REM
4606 REM CALLING PARAMETERS:
4607 REM FB$=NAME OF BACKUP FILE TO CREATE
4608 REM F$=NAME OF FILE TO DELETE FROM
4609 REM
4610 REM RETURNED PARAMETERS:
4611 REM L=-1 IF RECORD NOT FOUND
4612 REM
4613 REM COMMENT- USER IS ASKED FOR RECORD NUMBER
4614 REM BACKUP FILE IS CREATED
4615 REM
4616 REM ----------------------------
4620 N$="DELETE RECORD"
4630 GRAPHICS 0:PRINT N$:PRINT
4640 Q1$="NUMBER OF RECORD TO DELETE"
4650 Q2$="";Q3$=""
4660 GOSUB 5000:REM ASK QUESTION
4670 INPUT L:REM GET REC NUMBER
4680 GOSUB 3800:REM CREATE BACKUP
4690 TRAP 4880
4700 OPEN #1,4,0,FB$
4710 OPEN #2,8,0,F$
4715 IF L=1 THEN 4770
4720 FOR I=1 TO L-1
4730 INPUT #1;P1$,P2$,P3$,P4$,P5$,P6$,P7$,P8$
4740 PRINT #2;P1$;EL$;P2$;EL$;P3$;EL$;P4$
4750 PRINT #2;P5$;EL$;P6$;EL$;P7$;EL$;P8$
4760 NEXT I
4770 REM READ BUT DON'T WRITE RECORD TO DELETE
4780 INPUT #1;P1$,P2$,P3$,P4$,P5$,P6$,P7$,P8$
4790 REM READ AND WRITE TILL END OF FILE
4800 TRAP 4860
4810 FOR I=1 TO 4000
4820 INPUT #1;P1$,P2$,P3$,P4$,P5$,P6$,P7$,P8$
4830 PRINT #2;P1$;EL$;P2$;EL$;P3$;EL$;P4$
4840 PRINT #2;P5$;EL$;P6$;EL$;P7$;EL$;P8$
4850 NEXT I
4860 REM NORMAL RETURN
4870 CLOSE #1:CLOSE #2:RETURN
4880 REM ERROR RETURN
4890 CLOSE #1:CLOSE #2
4900 L=-1
4910 RETURN
```

Figure A.4: Delete Record Subroutine



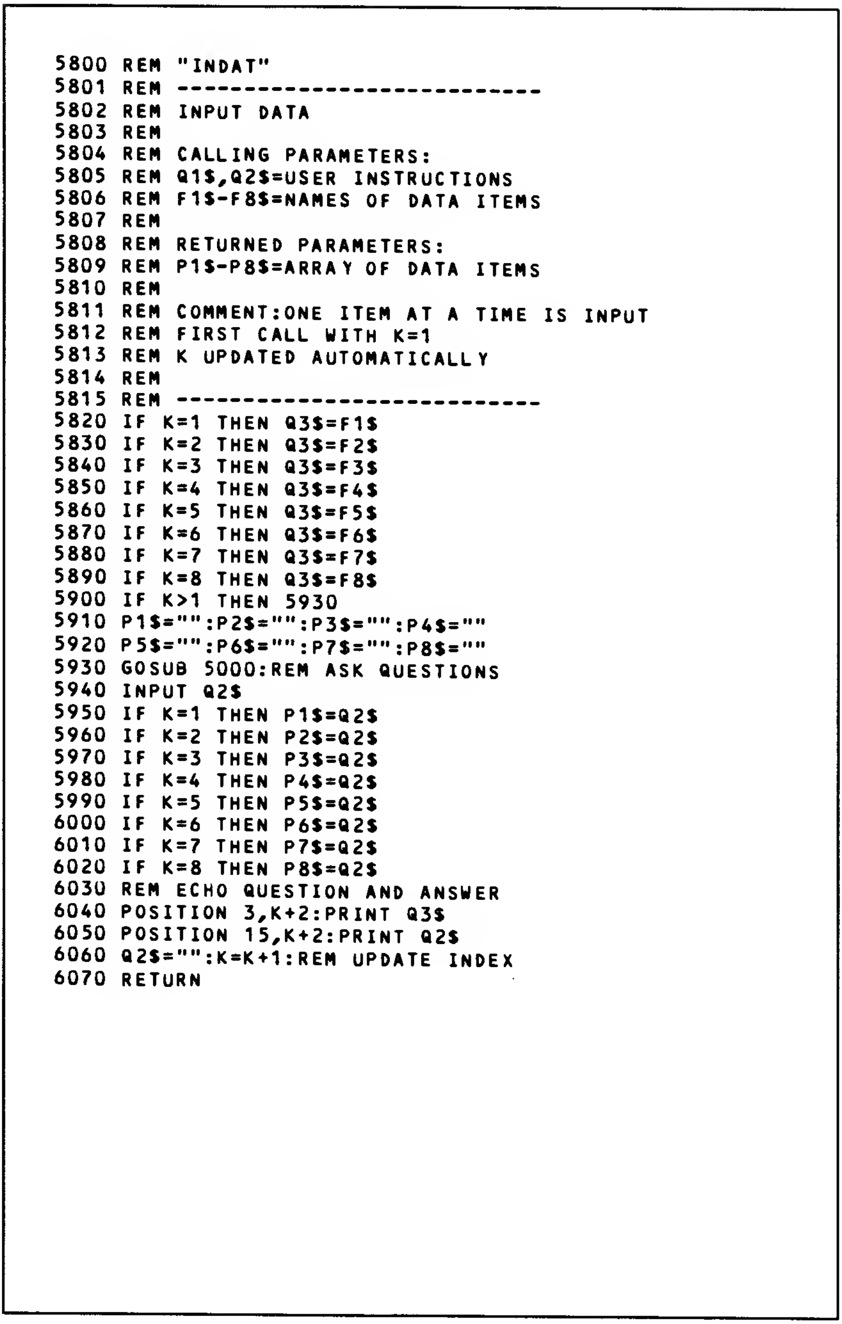
```
5000 REM "DIALOG"
5001 REM -----------------------------
5002 REM DIALOGUE
5003 REM
5004 REM CALLING PARAMETERS:
5005 REM Q1$.Q2$,Q3$=QUESTIONS TO DISPLAY
5006 REM
5007 REM RETURNED PARAMETERS:
5008 REM NONE
5009 REM
5010 REM COMMENT: UNUSED QUESTION STRINGS SHOULD
5011 REM BE SET TO NULL ( "" )
5012 REM
5013 REM -----------------------------
5015 POSITION 1,20:FOR I=1 TO 3:PRINT S$:NEXT I
5020 FOR I=19 TO 22
5030 POSITION 1,I
5040 IF I>19 THEN 5080
5050 FOR J=1 TO 38:REM DRAW A BORDER
5060 PRINT "-";
5070 NEXT J
5080 IF I=20 THEN ? Q1$:REM FIRST QUESTION
5090 IF I=21 THEN ? Q2$:REM SECOND QUESTION
5100 IF I=22 THEN ? Q3$;:REM THIRD QUESTION
5110 NEXT I
5120 RETURN
```

Figure A.5: Display Dialog Subroutine



```
5400 REM "INCH"
5401 REM -----------------------------
5402 REM INPUT CHARACTER
5403 REM
5404 REM CALLING PARAMETERS:
5405 REM NONE
5406 REM
5407 REM RETURNED PARAMETERS:
5408 REM CH$=INPUT CHARACTER
5409 REM
5410 REM -----------------------------
5420 REM WAIT FOR INPUT CHARACTER
5430 OPEN #3,4,0,"K:"
5440 GET #3,A
5450 CLOSE #3
5460 CH$=CHR$(A)
5470 RETURN
```

Figure A.6: Input Character Subroutine



```
5800 REM "INDAT"
5801 REM --------------------------
5802 REM INPUT DATA
5803 REM
5804 REM CALLING PARAMETERS:
5805 REM Q1$,Q2$=USER INSTRUCTIONS
5806 REM F1$-F8$=NAMES OF DATA ITEMS
5807 REM
5808 REM RETURNED PARAMETERS:
5809 REM P1$-P8$=ARRAY OF DATA ITEMS
5810 REM
5811 REM COMMENT:ONE ITEM AT A TIME IS INPUT
5812 REM FIRST CALL WITH K=1
5813 REM K UPDATED AUTOMATICALLY
5814 REM
5815 REM --------------------------
5820 IF K=1 THEN Q3$=F1$
5830 IF K=2 THEN Q3$=F2$
5840 IF K=3 THEN Q3$=F3$
5850 IF K=4 THEN Q3$=F4$
5860 IF K=5 THEN Q3$=F5$
5870 IF K=6 THEN Q3$=F6$
5880 IF K=7 THEN Q3$=F7$
5890 IF K=8 THEN Q3$=F8$
5900 IF K>1 THEN 5930
5910 P1$="":P2$="":P3$="":P4$=""
5920 P5$="":P6$="":P7$="":P8$=""
5930 GOSUB 5000:REM ASK QUESTIONS
5940 INPUT Q2$
5950 IF K=1 THEN P1$=Q2$
5960 IF K=2 THEN P2$=Q2$
5970 IF K=3 THEN P3$=Q2$
5980 IF K=4 THEN P4$=Q2$
5990 IF K=5 THEN P5$=Q2$
6000 IF K=6 THEN P6$=Q2$
6010 IF K=7 THEN P7$=Q2$
6020 IF K=8 THEN P8$=Q2$
6030 REM ECHO QUESTION AND ANSWER
6040 POSITION 3,K+2:PRINT Q3$
6050 POSITION 15,K+2:PRINT Q2$
6060 Q2$="":K=K+1:REM UPDATE INDEX
6070 RETURN
```

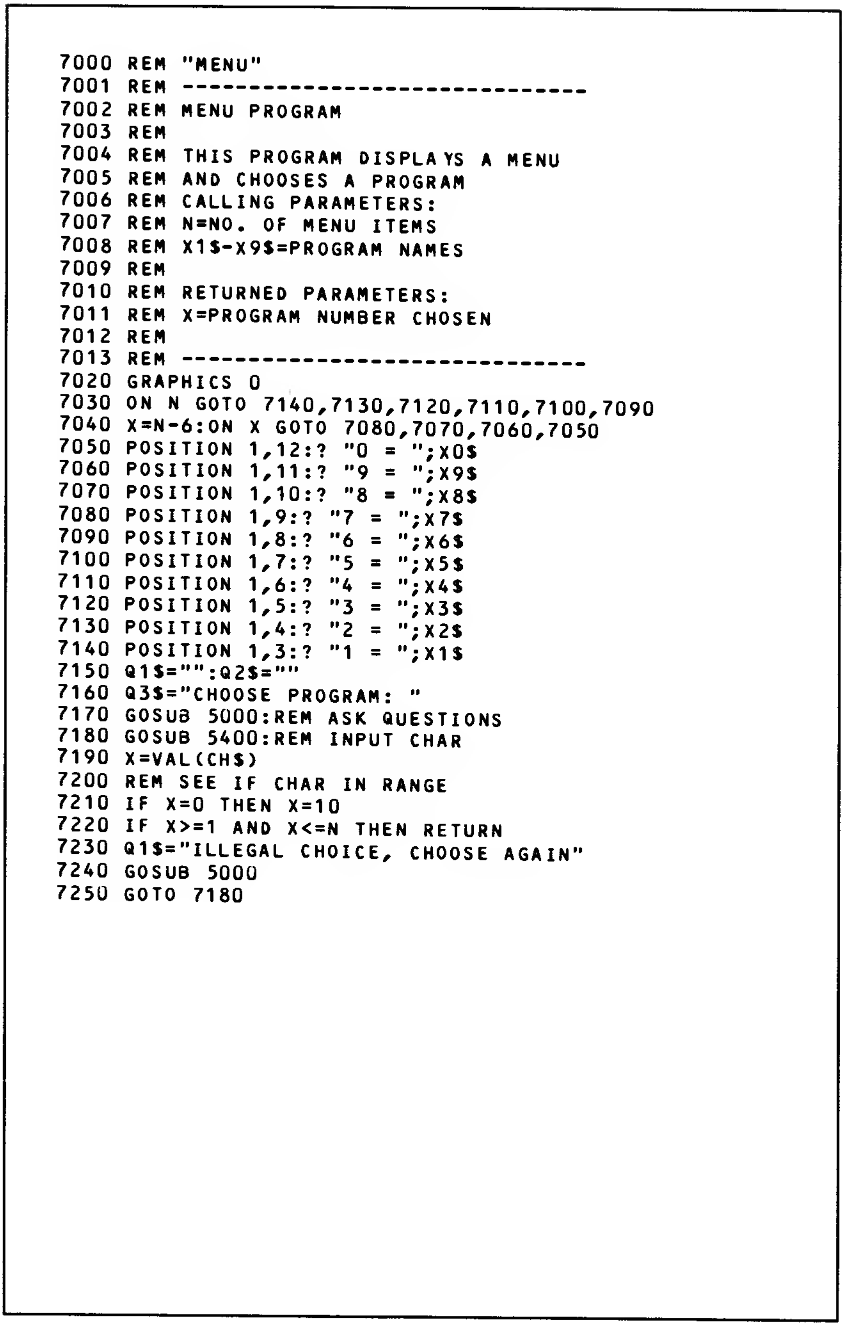
Figure A. 7: Input Data Subroutine

```
6200 REM "INIT"
6201 REM ------------------------------
6202 REM INITIALIZE DISPLAY
6203 REM
6204 REM THIS ROUTINE CLEARS THE SCREEN AND
6205 REM PRINTS THE PROGRAM TITLE
6206 REM CALLING PARAMETERS:
6207 REM N$=TITLE OF PROGRAM
6208 REM
6209 REM RETURNED PARAMETERS:
6210 REM NONE
6211 REM
6212 REM ------------------------------
6220 GRAPHICS 0:REM CLEAR SCREEN
6230 PRINT :PRINT
6240 DIM Q1$(40),Q2$(40),Q3$(40),CH$(2),N$(30)
6250 DIM YN$(2),X1$(38),X2$(38),X3$(38),X4$(38)
6260 DIM X5$(38),X6$(38),X7$(38),X8$(38),X9$(38)
6270 DIM X0$(38),P(10),S$(38)
6280 S$="                                      "
6290 DIM F1$(20),F2$(20),F3$(20),F4$(20)
6300 DIM F5$(20),F6$(2U),F7$(20),F8$(20)
6310 DIM P1$(20),P2$(20),P3$(20),P4$(20)
6320 DIM P5$(20),P6$(20),P7$(20),P8$(20)
6330 DIM F$(12),FB$(12),EL$(1)
6340 EL$=CHR$(155):REM END OF LINE CHAR
6350 RETURN
```

Figure A.8: Initialize Program and Display Subroutine

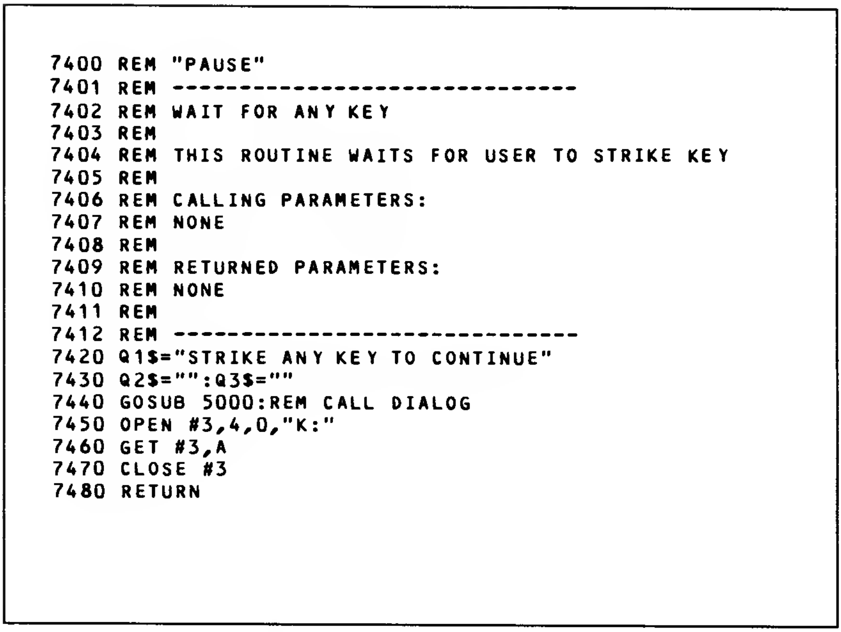
```
6600 REM "INPAR"
6601 REM ------------------------------
6602 REM INPUT PARAMETERS
6603 REM
6604 REM CALLING PARAMETERS:
6605 REM Q1$,Q2$=USER INSTRUCTIONS
6606 REM Q3$=NAME OF DATA ITEM
6607 REM
6608 REM RETURNED PARAMETERS:
6609 REM PAR(K)=THE DATA ITEM OBTAINED
6610 REM
6611 REM COMMENT: CALL WITH K=1 FOR FIRST DATA ITEM
6612 REM K AUTOMATICALLY UPDATED
6613 REM
6614 REM ------------------------------
6620 GOSUB 5000:REM ASK QUESTIONS
6630 INPUT X:P(K)=X:REM GET VALUE
6640 REM ECHO QUESTION AND ANSWER
6650 POSITION 3,K+2:PRINT Q3$
6660 POSITION 24,K+2:PRINT P(K)
6670 K=K+1
6680 RETURN
```

Figure A.9: Input Program Parameters Subroutine



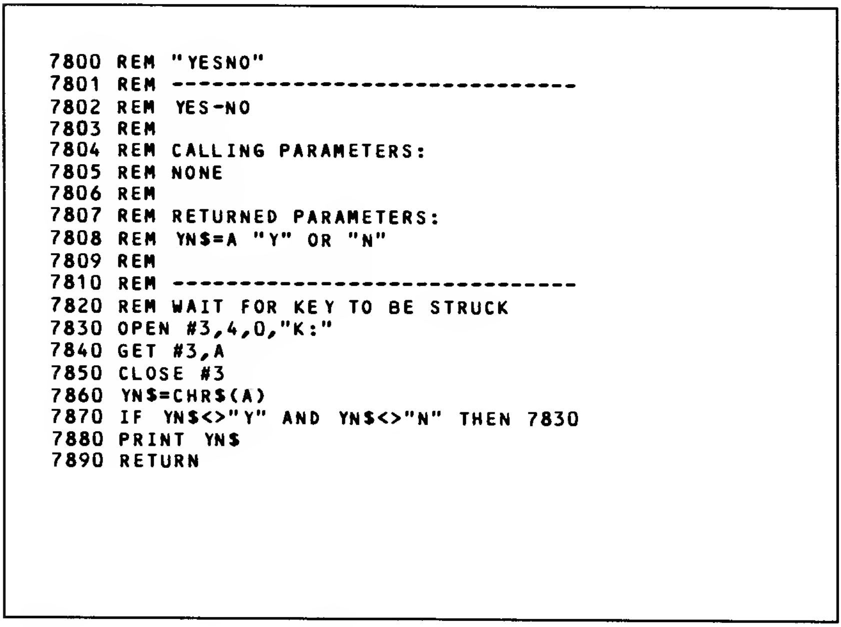
```
7000 REM "MENU"
7001 REM -----------------------------
7002 REM MENU PROGRAM
7003 REM
7004 REM THIS PROGRAM DISPLA YS A MENU
7005 REM AND CHOOSES A PROGRAM
7006 REM CALLING PARAMETERS:
7007 REM N=NO. OF MENU ITEMS
7008 REM X1$-X9$=PROGRAM NAMES
7009 REM
7010 REM RETURNED PARAMETERS:
7011 REM X=PROGRAM NUMBER CHOSEN
7012 REM
7013 REM -----------------------------
7020 GRAPHICS 0
7030 ON N GOTO 7140,7130,7120,7110,7100,7090
7040 X=N-6:ON X GOTO 7080,7070,7060,7050
7050 POSITION 1,12:? "0 = ";X0$
7060 POSITION 1,11:? "9 = ";X9$
7070 POSITION 1,10:? "8 = ";X8$
7080 POSITION 1,9:? "7 = ";X7$
7090 POSITION 1,8:? "6 = ";X6$
7100 POSITION 1,7:? "5 = ";X5$
7110 POSITION 1,6:? "4 = ";X4$
7120 POSITION 1,5:? "3 = ";X3$
7130 POSITION 1,4:? "2 = ";X2$
7140 POSITION 1,3:? "1 = ";X1$
7150 Q1$="":Q2$=""
7160 Q3$="CHOOSE PROGRAM: "
7170 GOSUB 5000:REM ASK QUESTIONS
7180 GOSUB 5400:REM INPUT CHAR
7190 X=VAL(CH$)
7200 REM SEE IF CHAR IN RANGE
7210 IF X=0 THEN X=10
7220 IF X>=1 AND X<=N THEN RETURN
7230 Q1$="ILLEGAL CHOICE, CHOOSE AGAIN"
7240 GOSUB 5000
7250 GOTO 7180
```

Figure A.10: Display Menu Subroutine



```
7400 REM "PAUSE"
7401 REM -----------------------------
7402 REM WAIT FOR ANY KEY
7403 REM
7404 REM THIS ROUTINE WAITS FOR USER TO STRIKE KEY
7405 REM
7406 REM CALLING PARAMETERS:
7407 REM NONE
7408 REM
7409 REM RETURNED PARAMETERS:
7410 REM NONE
7411 REM
7412 REM -----------------------------
7420 Q1$="STRIKE ANY KEY TO CONTINUE"
7430 Q2$="":Q3$=""
7440 GOSUB 5000:REM CALL DIALOG
7450 OPEN #3,4,0,"K:"
7460 GET #3,A
7470 CLOSE #3
7480 RETURN
```

Figure A.11: Pause Subroutine



```
7800 REM "YESNO"
7801 REM -----------------------------
7802 REM  YES-NO
7803 REM
7804 REM CALLING PARAMETERS:
7805 REM NONE
7806 REM
7807 REM RETURNED PARAMETERS:
7808 REM YN$=A "Y" OR "N"
7809 REM
7810 REM -----------------------------
7820 REM WAIT FOR KEY TO BE STRUCK
7830 OPEN #3,4,0,"K:"
7840 GET #3,A
7850 CLOSE #3
7860 YN$=CHR$(A)
7870 IF YN$<>"Y" AND YN$<>"N" THEN 7830
7880 PRINT YN$
7890 RETURN
```

Figure A.12: Yes or No Subroutine

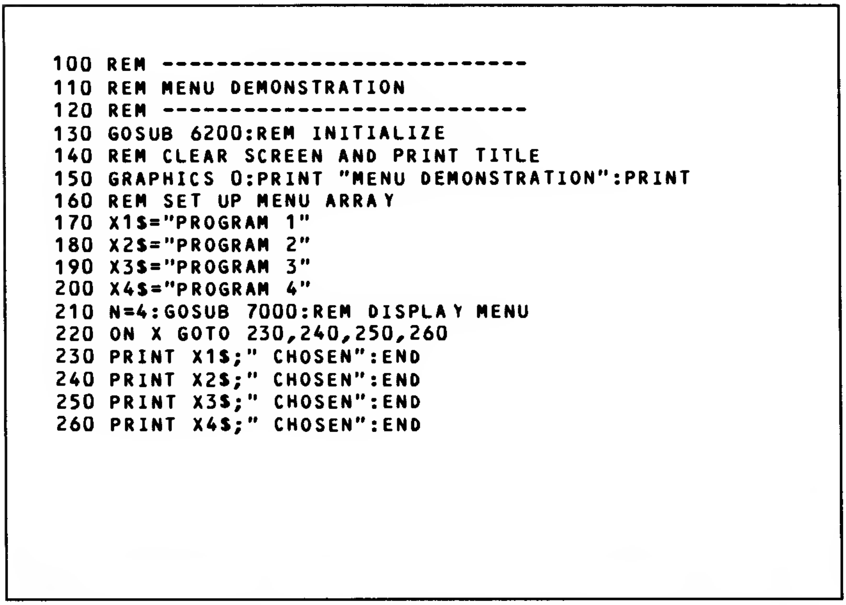
B
HOW TO USE THE CENTRAL SUBROUTINES

Readers who would like to improve their programming skills should try writing their own programs. You can save a lot of time and effort by using the central subroutines in this book as part of your programs. This appendix will help you learn to use these central subroutines. We review here several of the subroutines and how they function. We also include sample programs that show how the subroutines can be used.

MENU SET-UP

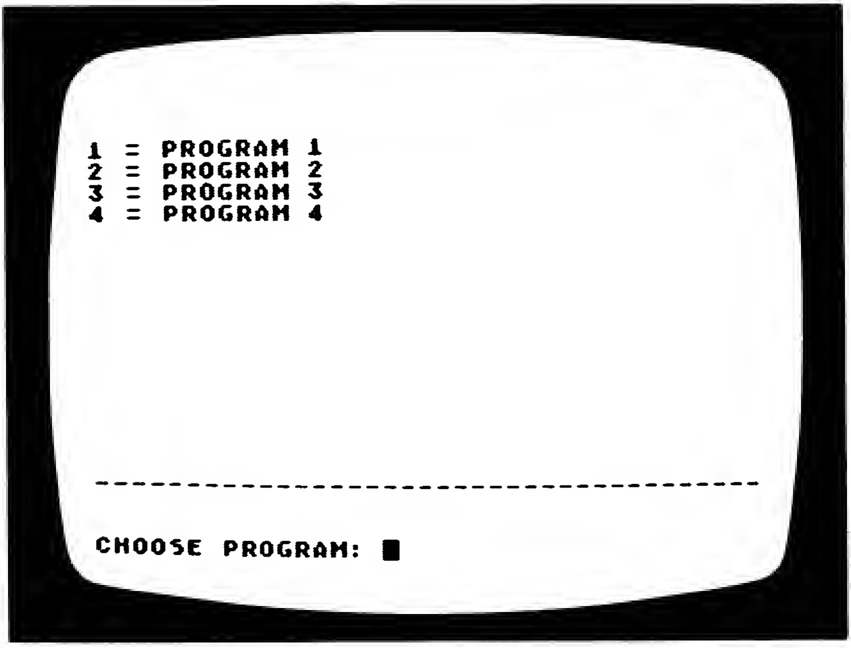
The menu set-up subroutine begins at line 7000. This subroutine displays a menu on the screen and waits for the user to press a number key. The menu program then returns a value, which you use to select the program you want to run.

Figures B.1 and B.2 show the program listing and screen display for a sample menu selection. We set up string variables with the names of the available programs. We then set N= to the number of program choices and call the menu subroutine. The subroutine displays the names of the programs and waits for the user to make a selection. The subroutine then returns the value of the selected program. The returned value is used in a GOSUB statement to select a program.



```
100 REM --------------------------
110 REM MENU DEMONSTRATION
120 REM --------------------------
130 GOSUB 6200:REM INITIALIZE
140 REM CLEAR SCREEN AND PRINT TITLE
150 GRAPHICS 0:PRINT "MENU DEMONSTRATION":PRINT
160 REM SET UP MENU ARRAY
170 X1$="PROGRAM 1"
180 X2$="PROGRAM 2"
190 X3$="PROGRAM 3"
200 X4$="PROGRAM 4"
210 N=4:GOSUB 7000:REM DISPLAY MENU
220 ON X GOTO 230,240,250,260
230 PRINT X1$;" CHOSEN":END
240 PRINT X2$;" CHOSEN":END
250 PRINT X3$;" CHOSEN":END
260 PRINT X4$;" CHOSEN":END
```

Figure B.1: Program Listing: Sample Menu Selection



```
1 = PROGRAM 1
2 = PROGRAM 2
3 = PROGRAM 3
4 = PROGRAM 4






--------------------------------------

CHOOSE PROGRAM: █
```

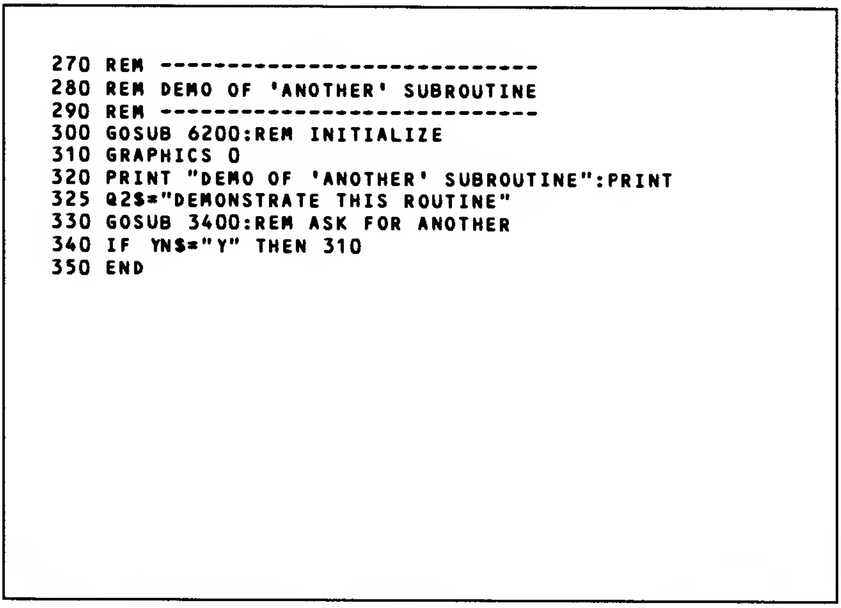
Figure B.2: Screen Display: Sample Menu Selection

CHOOSING ANOTHER PROGRAM

All the programs in this book use the ANOTH subroutine. This subroutine asks the user whether he would like to run the program again. The subroutine displays three lines of data. The ANOTH subroutine generates the first and third data lines. You specify the second line when you call the subroutine.

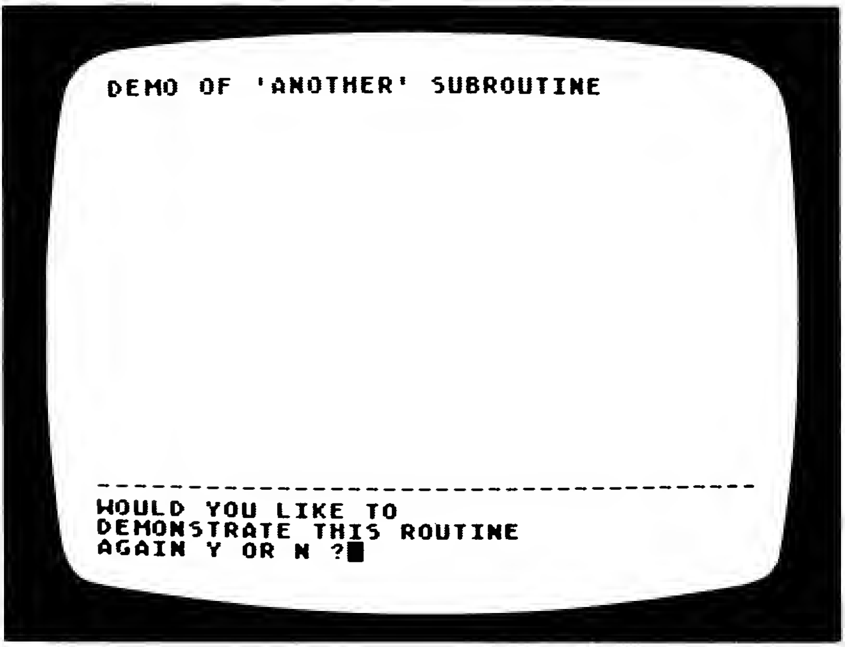
The ANOTH routine returns a Y or an N in the variable YN$. The program tests this variable to decide whether it will run the program again.

Figures B.3 and B.4 show the program listing and a screen display for the ANOTH demonstration. Note that the sample program displays the question DEMONSTRATE THIS ROUTINE.



```
270 REM ----------------------------
280 REM DEMO OF 'ANOTHER' SUBROUTINE
290 REM ----------------------------
300 GOSUB 6200:REM INITIALIZE
310 GRAPHICS 0
320 PRINT "DEMO OF 'ANOTHER' SUBROUTINE":PRINT
325 Q2$="DEMONSTRATE THIS ROUTINE"
330 GOSUB 3400:REM ASK FOR ANOTHER
340 IF YN$="Y" THEN 310
350 END
```

Figure B.3: Program Listing: Demonstration of ANOTH Subroutine



```
DEMO OF 'ANOTHER' SUBROUTINE









----------------------------------------
WOULD YOU LIKE TO
DEMONSTRATE THIS ROUTINE
AGAIN Y OR N ?█
```

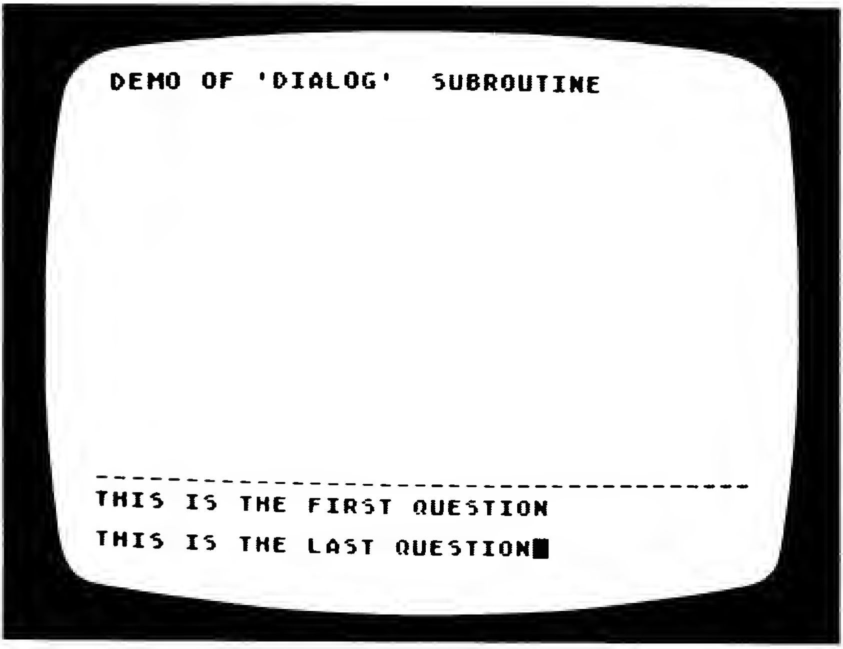
Figure B.4: Screen Display: Demonstration of ANOTH Subroutine

THE DIALOG SUBROUTINE

We use the DIALOG subroutine to display three lines of text at the bottom of the screen. You have the flexibility to generate three different questions; three questions or instructions will handle most programs you design. We often give an instruction on the first line, with questions on the second and third lines. This routine also draws a border and clears leftover text from previous questions. The routine exits with the cursor after the third question. Figures B.5 and B.6 show a program listing and screen display of a DIALOG demonstration.

```
360 REM ---------------------------
370 REM DEMO OF 'DIALOG' SUBROUTINE
380 REM ---------------------------
390 GOSUB 6200:REM INITIALIZE
400 GRAPHICS 0
410 PRINT "DEMO OF 'DIALOG'  SUBROUTINE"
420 PRINT
430 REM SET UP QUESTIONS
440 Q1$="THIS IS THE FIRST QUESTION"
450 Q2$="":REM A BLANK QUESTION
460 Q3$="THIS IS THE LAST QUESTION"
470 GOSUB 5000:REM SEND OUT QUESTIONS
475 FOR I=1 TO 2000:NEXT I
480 END
```

Figure B.5: Program Listing: Demonstration of DIALOG Subroutine



```
DEMO OF 'DIALOG'  SUBROUTINE









---------------------------------------
THIS IS THE FIRST QUESTION
THIS IS THE LAST QUESTION█
```

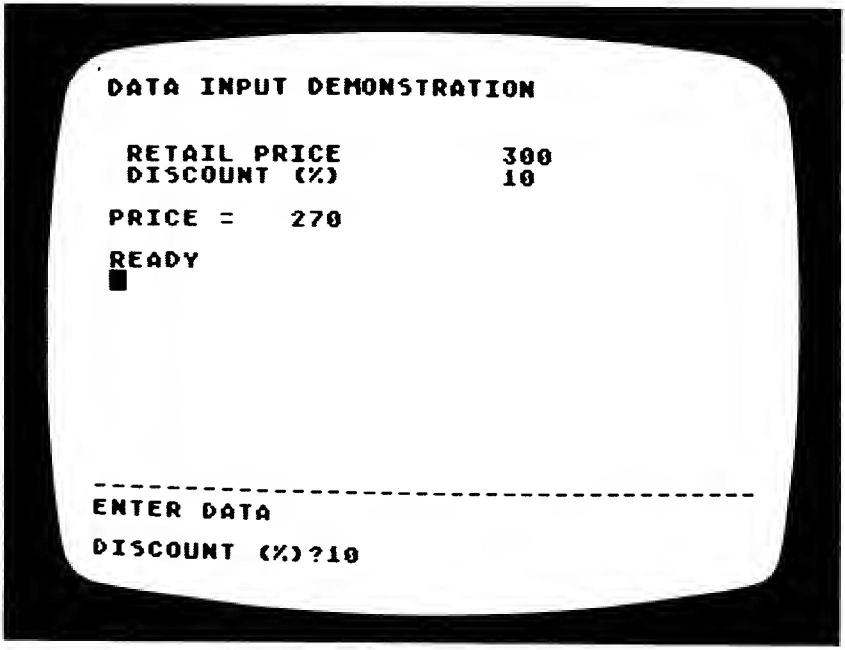
Figure B.6: Screen Display: Demonstration of DIALOG Subroutine

DATA INPUT SUBROUTINE

The INPAR program handles data input for many of the programs in this book and is easily used to create additional programs. You begin the data input process by clearing the screen with the INIT subroutine and setting the index K equal to 1. You are now ready to input program parameters. For each parameter, set Q3$ equal to the name of the parameter and call the subroutine at 6600. The INPAR subroutine asks the parameter name at the bottom of the screen and waits for input. The parameter then appears at the top of the screen. INPAR stores parameters in the array P(K) in the order of entry. Figures B.7 and B.8 show a program listing and screen display of the Data Input program.

```
490 REM -----------------------------
500 REM DATA INPUT DEMONSTRATION
510 REM -----------------------------
520 GOSUB 6200:REM INITIALIZE
530 GRAPHICS 0
540 PRINT "DATA INPUT DEMONSTRATION"
550 K=1
560 REM SET UP DIALOG AREA
570 Q1$="ENTER DATA":Q2$=""
580 Q3$="RETAIL PRICE":GOSUB 6600:REM GET FIRST ITEM
590 Q3$="DISCOUNT (%)":GOSUB 6600:REM SECOND ITEM
600 PRINT :PRINT "PRICE = ",
610 P=P(1)-P(2)/100*P(1)
620 P=INT(P*100+0.5)/100:REM ROUND OFF TO 2 DIGITS
630 PRINT P
640 END
```

Figure B. 7: Program Listing: Data Input Demonstration



```
DATA INPUT DEMONSTRATION

 RETAIL PRICE          300
 DISCOUNT (%)          10

PRICE =    270

READY
█




----------------------------------------
ENTER DATA

DISCOUNT (%)?10
```

Figure B. 8: Screen Display: Data Input Demonstration

The SYBEX Library

Buyer's Guide

THE BEST OF TI 99/4A™ CARTRIDGES

by Thomas Blackadar

150 pp., illustr., Ref. 0-137

Save yourself time and frustration when buying TI 99/4A software. This buyer's guide gives an overview of the best available programs, with information on how to set up the computer to run them.

FAMILY COMPUTERS UNDER $200

by Doug Mosher

160 pp., illustr., Ref. 0-149

Find out what these inexpensive machines can do for you and your family. "If you're just getting started . . . this is the book to read before you buy."—Richard O'Reilly, Los Angeles newspaper columnist

PORTABLE COMPUTERS

by Sheldon Crop and Doug Mosher

128 pp., illustr., Ref. 0-144

"This book provides a clear and concise introduction to the expanding new world of personal computers."—Mark Powelson, Editor, San Francisco Focus Magazine

THE BEST OF VIC-20™ SOFTWARE

by Thomas Blackadar

150 pp., illustr., Ref. 0-139

Save yourself time and frustration with this buyer's guide to VIC-20 software. Find the best game, music, education, and home management programs on the market today.

SELECTING THE RIGHT DATA BASE SOFTWARE

SELECTING THE RIGHT WORD PROCESSING SOFTWARE

SELECTING THE RIGHT SPREADSHEET SOFTWARE

by Kathy McHugh and Veronica Corchado

80 pp., illustr., Ref. 0-174, 0-177, 0-178

This series on selecting the right business software offers the busy professional concise, informative reviews of the best available software packages.

Introduction to Computers

OVERCOMING COMPUTER FEAR

by Jeff Berner

112 pp., illustr., Ref. 0-145

This easy-going introduction to computers helps you separate the facts from the myths.

COMPUTER ABC'S

by Daniel Le Noury and Rodnay Zaks

64 pp., illustr., Ref. 0-167

This beautifully illustrated, colorful book for parents and children takes you alphabetically through the world of computers, explaining each concept in simple language.

PARENTS, KIDS, AND COMPUTERS

by Lynn Alpers and Meg Holmberg

208 pp., illustr., Ref. 0-151

This book answers your questions about the educational possibilities of home computers.

THE COLLEGE STUDENT'S COMPUTER HANDBOOK

by Bryan Pfaffenberger

350 pp., illustr., Ref. 0-170

This friendly guide will aid students in selecting a computer system for college study, managing information in a college course, and writing research papers.

COMPUTER CRAZY

by Daniel Le Noury

100 pp., illustr., Ref. 0-173

No matter how you feel about computers, these cartoons will have you laughing about them.

DON'T! (or How to Care for Your Computer)

by Rodnay Zaks

214pp., 100 illustr., Ref. 0-065

The correct way to handle and care for all elements of a computer system, including what to do when something doesn't work.

YOUR FIRST COMPUTER

by Rodnay Zaks

258 pp., 150 illustr., Ref. 0-045

The most popular introduction to small computers and their peripherals: what they do and how to buy one.

INTERNATIONAL MICROCOMPUTER DICTIONARY

120 pp., Ref. 0-067

All the definitions and acronyms of microcomputer jargon defined in a handy pocket-sized edition. Includes translations of the most popular terms into ten languages.

FROM CHIPS TO SYSTEMS: AN INTRODUCTION TO MICROPROCESSORS

by Rodnay Zaks

552 pp., 400 illustr., Ref. 0-063

A simple and comprehensive introduction to microprocessors from both a hardware and software standpoint: what they are, how they operate, how to assemble them into a complete system.

Personal Computers

ATARI

YOUR FIRST ATARI® PROGRAM

by Rodnay Zaks

150 pp., illustr., Ref. 0-130

A fully illustrated, easy-to-use introduction to ATARI BASIC programming. Will have the reader programming in a matter of hours.

BASIC EXERCISES FOR THE ATARI®

by J.P. Lamoitier

251 pp., illustr., Ref. 0-101

Teaches ATARI BASIC through actual practice using graduated exercises drawn from everyday applications.

THE EASY GUIDE TO YOUR ATARI® 600XL/800XL

by Thomas Blackadar

175 pp., illustr., Ref. 0-125

This jargon-free companion will help you get started on the right foot with your new 600XL or 800XL ATARI computer.

ATARI® BASIC PROGRAMS IN MINUTES

by Stanley R. Trost

170 pp., illustr., Ref. 0-143

You can use this practical set of programs without any prior knowledge of BASIC! Application examples are taken from a wide variety of fields, including business, home management, and real estate.

Commodore 64/VIC-20

THE COMMODORE 64™/VIC-20™ BASIC HANDBOOK

by Douglas Hergert

144 pp., illustr., Ref. 0-116

A complete listing with descriptions and instructive examples of each of the Commodore 64 BASIC keywords and functions. A handy reference guide, organized like a dictionary.

THE EASY GUIDE TO YOUR COMMODORE 64™

by Joseph Kascmer

160 pp., illustr., Ref. 0-129

A friendly introduction to using the Commodore 64.

YOUR FIRST VIC-20™ PROGRAM

by Rodnay Zaks

150 pp., illustr., Ref. 0-129

A fully illustrated, easy-to-use introduction to VIC-20 BASIC programming. Will have the reader programming in a matter of hours.

THE VIC-20™ CONNECTION

by James W. Coffron

260 pp., 120 illustr., Ref. 0-128

Teaches elementary interfacing and BASIC programming of the VIC-20 for connection to external devices and household appliances.

YOUR FIRST COMMODORE 64™ PROGRAM

by Rodnay Zaks

182 pp., illustr., Ref. 0-172

You can learn to write simple programs without any prior knowledge of mathematics or computers! Guided by colorful illustrations and step-by-step instructions, you'll be constructing programs within an hour or two.

COMMODORE 64™ BASIC PROGRAMS IN MINUTES

by Stanley R. Trost

170 pp., illustr., Ref. 0-154

Here is a practical set of programs for business, finance, real estate, data analysis, record keeping and educational applications.

GRAPHICS GUIDE TO THE COMMODORE 64™

by Charles Platt

192 pp., illustr., Ref. 0-138

This easy-to-understand book will appeal to anyone who wants to master the Commodore 64's powerful graphics features.

IBM

THE ABC'S OF THE IBM® PC

by Joan Lasselle and Carol Ramsay

100 pp., illustr., Ref. 0-102

This is the book that will take you through the first crucial steps in learning to use the IBM PC.

THE BEST OF IBM® PC SOFTWARE

by Stanley R. Trost

144 pp., illustr., Ref. 0-104

Separates the wheat from the chaff in the world of IBM PC software. Tells you what to expect from the best available IBM PC programs.

THE IBM® PC-DOS HANDBOOK

by Richard Allen King

144 pp., illustr., Ref. 0-103

Explains the PC disk operating system, giving the user better control over the system. Get the most out of your PC by adapting its capabilities to your specific needs.

BUSINESS GRAPHICS FOR THE IBM® PC

by Nelson Ford

200 pp., illustr., Ref. 0-124

Ready-to-run programs for creating line graphs, complex illustrative multiple bar graphs, picture graphs, and more. An ideal way to use your PC's business capabilities!

THE IBM® PC CONNECTION

by James W. Coffron

200 pp., illustr., Ref. 0-127

Teaches elementary interfacing and BASIC programming of the IBM PC for connection to external devices and household appliances.

BASIC EXERCISES FOR THE IBM® PERSONAL COMPUTER

by J.P. Lamoitier

252 pp., 90 illustr., Ref. 0-088

Teaches IBM BASIC through actual practice, using graduated exercises drawn from everyday applications.

USEFUL BASIC PROGRAMS FOR THE IBM® PC

by Stanley R. Trost

144 pp., Ref. 0-111

This collection of programs takes full advantage of the interactive capabilities of your IBM Personal Computer. Financial calculations, investment analysis, record keeping, and math practice—made easier on your IBM PC.

YOUR FIRST IBM® PC PROGRAM

by Rodnay Zaks

182 pp., illustr., Ref. 0-171

This well-illustrated book makes programming easy for children and adults.

YOUR IBM® PC JUNIOR

by Douglas Hergert

250 pp., illustr., Ref. 0-179

This comprehensive reference guide to IBM's most economical microcomputer offers many practical applications and all the helpful information you'll need to get started with your IBM PC Junior.

DATA FILE PROGRAMMING ON YOUR IBM® PC

by Alan Simpson

275 pp., illustr., Ref. 0-146

This book provides instructions and examples of managing data files in BASIC. Programming designs and developments are extensively discussed.

Apple

THE EASY GUIDE TO YOUR APPLE II®

by Joseph Kascmer

160 pp., illustr., Ref. 0-122

A friendly introduction to using the Apple II, II plus and the new IIe.

BASIC EXERCISES FOR THE APPLE®

by J.P. Lamoitier

250 pp., 90 illustr., Ref. 0-084

Teaches Apple BASIC through actual practice, using graduated exercises drawn from everyday applications.

APPLE II® BASIC HANDBOOK

by Douglas Hergert

144 pp., illustr., Ref. 0-155

A complete listing with descriptions and instructive examples of each of the Apple II BASIC keywords and functions. A handy reference guide, organized like a dictionary.

APPLE II® BASIC PROGRAMS IN MINUTES

by Stanley R. Trost

150 pp., illustr., Ref. 0-121

A collection of ready-to-run programs for financial calculations, investment analysis, record keeping, and many more home and office applications. These programs can be entered on your Apple II plus or IIe in minutes!

YOUR FIRST APPLE II® PROGRAM

by Rodnay Zaks

150 pp., illustr., Ref. 0-136

A fully illustrated, easy-to-use introduction to APPLE BASIC programming. Will have the reader programming in a matter of hours.

THE APPLE® CONNECTION

by James W. Coffron

264 pp., 120 illustr., Ref. 0-085

Teaches elementary interfacing and BASIC programming of the Apple for connection to external devices and household appliances.

TRS-80

YOUR COLOR COMPUTER
by Doug Mosher
350 pp., illustr., Ref. 0-097
Patience and humor guide the reader through purchasing, setting up, programming, and using the Radio Shack TRS-80/TDP Series 100 Color Computer. A complete introduction.

THE FOOLPROOF GUIDE TO SCRIPSIT™ WORD PROCESSING
by Jeff Berner
225 pp., illustr., Ref. 0-098
Everything you need to know about SCRIPSIT—from starting out, to mastering document editing. This user-friendly guide is written in plain English, with a touch of wit.

Other Popular Computers

YOUR FIRST TI 99/4A™ PROGRAM
by Rodnay Zaks
182 pp., illustr., Ref. 0-157
Colorfully illustrated, this book concentrates on the essentials of programming in a clear, entertaining fashion.

THE RADIO SHACK® NOTEBOOK COMPUTER
by Orson Kellogg
128 pp., illustr., Ref. 0-150
Whether you already have the Radio Shack Model 100 notebook computer, or are interested in buying one, this book will clearly explain what it can do for you.

THE EASY GUIDE TO YOUR COLECO ADAM™
by Thomas Blackadar
175 pp., illustr., Ref. 0-181
This quick reference guide shows you how to get started on your Coleco Adam with a minimum of technical jargon.

YOUR KAYPRO II/4/10™
by Andrea Reid and Gary Deidrichs
250 pp., illustr., Ref. 0-166
This book is a non-technical introduction to the KAYPRO family of computers. You will find all you need to know about operating your KAYPRO within this one complete guide.

Software and Applications

Operating Systems

THE CP/M® HANDBOOK
by Rodnay Zaks
320 pp., 100 illustr., Ref 0-048
An indispensable reference and guide to CP/M—the most widely-used operating system for small computers.

MASTERING CP/M®
by Alan R. Miller
398 pp., illustr., Ref. 0-068
For advanced CP/M users or systems programmers who want maximum use of the CP/M operating system . . . takes up where our *CP/M Handbook* leaves off.

THE BEST OF CP/M® SOFTWARE
by John D. Halamka
250 pp., illustr., Ref. 0-100
This book reviews tried-and-tested, commercially available software for your CP/M system.

REAL WORLD UNIX
by John D. Halamka
250 pp., illustr., Ref. 0-093
This book is written for the beginning and intermediate UNIX user in a practical, straightforward manner, with specific instructions given for many special applications.

THE CP/M PLUS™ HANDBOOK

by Alan R. Miller

250 pp., illustr., Ref. 0-158

This guide is easy for the beginner to understand, yet contains valuable information for advanced users of CP/M Plus (Version 3).

Business Software

INTRODUCTION TO WORDSTAR™

by Arthur Naiman

202 pp., 30 illustr., Ref. 0-077

Makes it easy to learn how to use WordStar, a powerful word processing program for personal computers.

PRACTICAL WORDSTAR™ USES

by Julie Anne Arca

200 pp., illustr., Ref. 0-107

Pick your most time-consuming office tasks and this book will show you how to streamline them with WordStar.

MASTERING VISICALC®

by Douglas Hergert

217 pp., 140 illustr., Ref. 0-090

Explains how to use the VisiCalc "electronic spreadsheet" functions and provides examples of each. Makes using this powerful program simple.

DOING BUSINESS WITH VISICALC®

by Stanley R. Trost

260 pp., Ref. 0-086

Presents accounting and management planning applications—from financial statements to master budgets; from pricing models to investment strategies.

DOING BUSINESS WITH SUPERCALC™

by Stanley R. Trost

248 pp., illustr., Ref. 0-095

Presents accounting and management planning applications—from financial statements to master budgets; from pricing models to investment strategies.

VISICALC® FOR SCIENCE AND ENGINEERING

by Stanley R. Trost and Charles Pomernackl

225 pp., illustr., Ref. 0-096

More than 50 programs for solving technical problems in the science and engineering fields. Applications range from math and statistics to electrical and electronic engineering.

DOING BUSINESS WITH 1-2-3™

by Stanley R. Trost

250 pp., illustr., Ref. 0-159

If you are a business professional using the 1-2-3 software package, you will find the spreadsheet and graphics models provided in this book easy to use "as is" in everyday business situations.

THE ABC'S OF 1-2-3™

by Chris Gilbert

225 pp., illustr., Ref. 0-168

For those new to the LOTUS 1-2-3 program, this book offers step-by-step instructions in mastering its spreadsheet, data base, and graphing capabilities.

UNDERSTANDING dBASE II™

by Alan Simpson

220 pp., illustr., Ref. 0-147

Learn programming techniques for mailing label systems, bookkeeping and data base management, as well as ways to interface dBASE II with other software systems.

DOING BUSINESS WITH dBASE II™

by Stanley R. Trost

250 pp., illustr., Ref. 0-160

Learn to use dBASE II for accounts receivable, recording business income and expenses, keeping personal records and mailing lists, and much more.

DOING BUSINESS WITH MULTIPLAN

by Richard Allen King and Stanley R. Trost

250 pp., illustr., Ref. 0-148

This book will show you how using Multiplan can be nearly as easy as learning to use a pocket calculator. It presents a collection of templates that can be applied "as is" to business situations.

DOING BUSINESS WITH PFS®

by Stanley R. Trost

250 pp., illustr., Ref. 0-161

This practical guide describes specific business and personal applications in detail. Learn to use PFS for accounting, data analysis, mailing lists and more.

INFOPOWER: PRACTICAL INFOSTAR USES

by Jule Anne Arca and Charles F. Pirro

275 pp., illustr., Ref. 0-108

This book gives you an overview of InfoStar, including DataStar and ReportStar, WordStar, MailMerge, and SuperSort. Hands on exercises take you step-by-step through real life business applications.

WRITING WITH EASYWRITER II™

by Douglas W. Topham

250 pp., illustr., Ref. 0-141

Friendly style, handy illustrations, and numerous sample exercises make it easy to learn the EasyWriter II word processing system.

Business Applications

INTRODUCTION TO WORD PROCESSING

by Hal Glatzer

205 pp., 140 illustr., Ref. 0-076

Explains in plain language what a word processor can do, how it improves productivity, how to use a word processor and how to buy one wisely.

COMPUTER POWER FOR YOUR LAW OFFICE

by Daniel Remer

225 pp., Ref. 0-109

How to use computers to reach peak productivity in your law office, simply and inexpensively.

OFFICE EFFICIENCY WITH PERSONAL COMPUTERS

by Sheldon Crop

175 pp., illustr., Ref. 0-165

Planning for computerization of your office? This book provides a simplified discussion of the challenges involved for everyone from business owner to clerical worker.

COMPUTER POWER FOR YOUR ACCOUNTING OFFICE

by James Morgan

250 pp., illustr., Ref. 0-164

This book is a convenient source of information about computerizing you accounting office, with an emphasis on hardware and software options.

BASIC

YOUR FIRST BASIC PROGRAM

by Rodnay Zaks

150pp. illustr. in color, Ref. 0-129

A "how-to-program" book for the first time computer user, aged 8 to 88.

FIFTY BASIC EXERCISES

by J. P. Lamoitier

232 pp., 90 illustr., Ref. 0-056

Teaches BASIC by actual practice, using graduated exercises drawn from everyday applications. All programs written in Microsoft BASIC.

INSIDE BASIC GAMES

by Richard Mateosian

348 pp., 120 illustr., Ref. 0-055

Teaches interactive BASIC programming through games. Games are written in Microsoft BASIC and can run on the TRS-80, Apple II and PET/CBM.

BASIC FOR BUSINESS

by Douglas Hergert

224 pp., 15 illustr., Ref. 0-080

A logically organized, no-nonsense introduction to BASIC programming for business applications. Includes many fully-explained accounting programs, and shows you how to write them.

EXECUTIVE PLANNING WITH BASIC

by X. T. Bui

196 pp., 19 illustr., Ref. 0-083

An important collection of business management decision models in BASIC, including Inventory Management (EOQ), Critical Path Analysis and PERT, Financial Ratio Analysis, Portfolio Management, and much more.

BASIC PROGRAMS FOR SCIENTISTS AND ENGINEERS

by Alan R. Miller

318 pp., 120 illustr., Ref. 0-073

This book from the "Programs for Scientists and Engineers" series provides a library of problem-solving programs while developing proficiency in BASIC.

CELESTIAL BASIC

by Eric Burgess

300 pp., 65 illustr., Ref. 0-087

A collection of BASIC programs that rapidly complete the chores of typical astronomical computations. It's like having a planetarium in your own home! Displays apparent movement of stars, planets and meteor showers.

YOUR SECOND BASIC PROGRAM

by Gary Lippman

250 pp., illustr., Ref. 0-152

A sequel to *Your First BASIC Program*, this book follows the same patient, detailed approach and brings you to the next level of programming skill.

Assembly Language Programming

PROGRAMMING THE 6502

by Rodnay Zaks

386 pp., 160 illustr., Ref. 0-046

Assembly language programming for the 6502, from basic concepts to advanced data structures.

6502 APPLICATIONS

by Rodnay Zaks

278 pp., 200 illustr., Ref. 0-015

Real-life application techniques: the input/output book for the 6502.

ADVANCED 6502 PROGRAMMING

by Rodnay Zaks

292 pp., 140 illustr., Ref. 0-089

Third in the 6502 series. Teaches more advanced programming techniques, using games as a framework for learning.

Hardware and Peripherals

MICROPROCESSOR INTERFACING TECHNIQUES

by Rodnay Zaks and Austin Lesea

456 pp., 400 illustr., Ref. 0-029

Complete hardware and software interconnect techniques, including D to A conversion, peripherals, standard buses and troubleshooting.

THE RS-232 SOLUTION

by Joe Campbell

225 pp., illustr., Ref. 0-140

Finally, a book that will show you how to correctly interface your computer to any RS-232-C peripheral.

USING CASSETTE RECORDERS WITH COMPUTERS

by James Richard Cook

175 pp., illustr., Ref. 0-169

Whatever your computer or application, you will find this book helpful in explaining details of cassette care and maintenance.

For a complete catalog of our publications:

U.S.A.
SYBEX, Inc.
2344 Sixth Street
Berkeley,
California 94710
Tel: (800) 227-2346
(415) 848-8233
Telex: 336311

FRANCE
SYBEX
4 Place Félix-Eboué
75583 Paris Cedex 12
France
Tel: 1/347-30-20
Telex: 211801

GERMANY
SYBEX-VERLAG
Heyestr. 22
4000 Düsseldorf 12
West Germany
Tel: (0211) 287066
Telex: 08 588 163

ATARI
BASIC PROGRAMS IN MINUTES

You don't have to be a programmer to program your Atari!

These versatile, ready-to-enter programs will enable you to perform over 65 home and business tasks with your Atari 800 or any of the new Atari XL series with 48K of memory.

Use these handy programs for:

- **Home Finances—**
  Future Value of an IRA
  Building a College Fund
- **Business Calculations—**
  Break-Even Point
  Depreciation Schedules
- **Real Estate—**
  Affordable Home Price
  Effect of Accelerating Mortgage Payments
- **Data Analysis—**
  Input and Plot Data
  Compute Moving Averages
- **Record Keeping—**
  Telephone and Mailing Lists
  Automobile Log
- **Education—**
  Arithmetic Drills

You will be surprised at how easy it is to enter and use these programs! Many of them can be entered and ready to run in less than ten minutes. No knowledge of BASIC programming is necessary to use this collection of useful programs.

ABOUT THE AUTHOR:

Stanley R. Trost is a principal in the firm SR Trost & Associates, Walnut Creek, CA. He consults on the use of computer systems for business, medical, and engineering applications. He earned a Master's degree in Electronic Engineering, as well as an MBA. Mr. Trost is the author of several books, including *Doing Business with VisiCalc, Doing Business with SuperCalc, VisiCalc for Science and Engineering,* and *Useful BASIC Programs for the IBM PC.*

00143
0 25211 01295

ISBN 0-89588-143-8